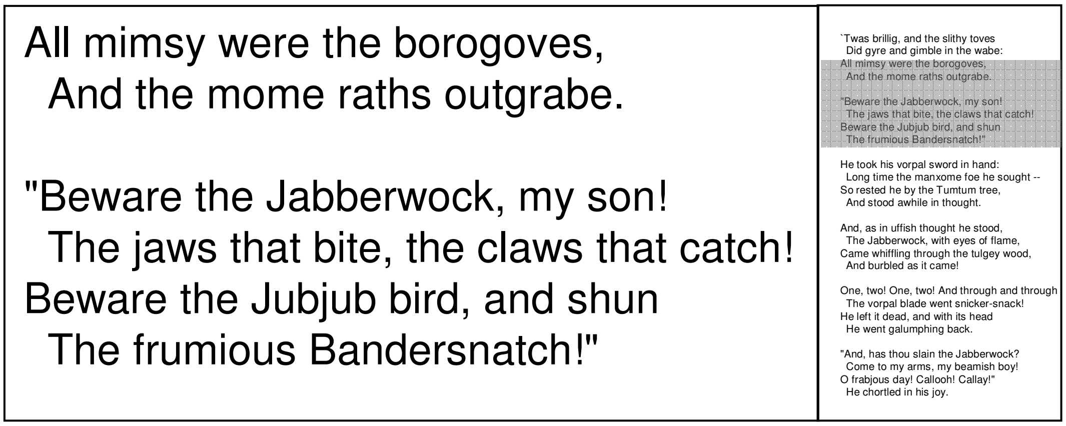
6.831 User Interface Design & Implementation
Fall 2006

Quiz 2

This quiz is closed book, closed notes. You have 80 minutes to complete it. Each question has enough space for a complete, concise answer, but an additional blank page is included at the end if you find it necessary to continue your answer.

Your name: ____

Suppose you're implementing a new kind of scrolling pane widget. Instead of displaying a conventional scrollbar for moving around its child component, the widget should display a miniature view of the entire child, with a **25% gray** rectangle showing which part of the child is currently visible. The example below shows how the scrolling widget might look with a text widget as its child. The input to the miniature view is like a scrollbar, so (for example) clicking and dragging the gray rectangle should behave the same as clicking and dragging a scrollbar thumb.



All mimsy were the borogoves,
  And the mome raths outgrabe.

"Beware the Jabberwock, my son!
  The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
  The frumious Bandersnatch!"

`Twas brillig, and the slithy toves
  Did gyre and gimble in the wabe:
All mimsy were the borogoves,
  And the mome raths outgrabe.

"Beware the Jabberwock, my son!
  The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
  The frumious Bandersnatch!"

He took his vorpal sword in hand:
  Long time the manxome foe he sought --
So rested he by the Tumtum tree,
  And stood awhile in thought.

And, as in uffish thought he stood,
  The Jabberwock, with eyes of flame,
Came whiffling through the tulgey wood,
  And burbled as it came!

One, two! One, two! And through and through
  The vorpal blade went snicker-snack!
He left it dead, and with its head
  He went galumphing back.

"And, has thou slain the Jabberwock?
  Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!"
  He chortled in his joy.

All the parts of the widget should be implemented as components, named as follows:

- **MiniScrollPane**: the entire widget itself
- **Viewport**: the part that shows the child component at normal size, on the left
- **Track**: the miniature view of the child component, on the right
- **Thumb**: the gray rectangle
- **Child**: the child component being scrolled (which happens to be a text editor in the example above, but might be another component).

1. Draw the component hierarchy showing all five components of the MiniScrollPane widget.

```
MiniScrollPane -> Viewport -> Child
               -> Track -> Thumb
```

The child can't be in two places in a component hierarchy – remember that the component hierarchy is a ***tree****. It should be a child of the viewport so that the viewport clips it.*

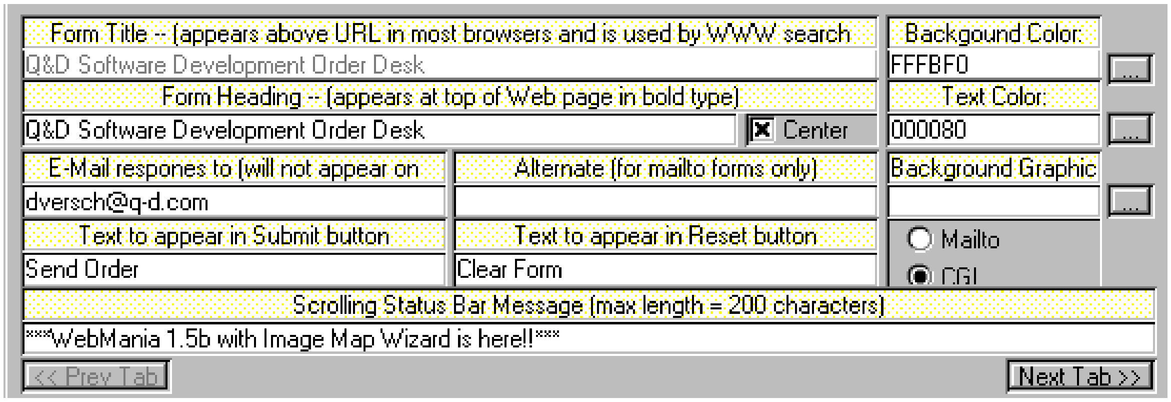
2. Write constraint equations that specify the vertical position and size of **Thumb** in terms of the positions and sizes of the other components.

Thumb.y / Track.height = -Child.y / Child.height
Thumb.height / Track.height = Viewport.height / Child.height

Note that we use "-Child.y" because Child.y is relative to the coordinate system of its parent, viewport. For example, if the child is scrolled down 100 pixels, that actually means that its y position is -100.

3. Fill in the paintComponent methods for **Track** and **Thumb** below. (In Swing, paintComponent() paints just the component itself; any children of the component are automatically painted next.)

```
// references to other components
JComponent miniScrollPane, viewport, child, track, thumb;

class Track extends JComponent {
   public void paintComponent(Graphics g) {
      Graphics2D g2 = (Graphics2D) g;

      scale = track.height / child.height;
      g2.scale(scale, scale);
      child.paint(g);
   }
}

class Thumb extends JComponent {
   public void paintComponent(Graphics g) {
      Graphics2D g2 = (Graphics2D) g;

      g2.setColor(new Color(0, 0, 0, 0.25));
      g2.fillRect(0, 0, thumb.width, thumb.height);
   }
}
```



Form Title -- (appears above URL in most browsers and is used by WWW search

Q&D Software Development Order Desk

Backgound Color:

FFFBF0

Form Heading -- (appears at top of Web page in bold type)

Q&D Software Development Order Desk ☒ Center

Text Color:

000080

E-Mail respones to (will not appear on

dversch@q-d.com

Alternate (for mailto forms only)

Background Graphic

Text to appear in Submit button

Send Order

Text to appear in Reset button

Clear Form

○ Mailto

● CGI

Scrolling Status Bar Message (max length = 200 characters)

WebMania 1.5b with Image Map Wizard is here!!

4. List at least 3 different violations of **graphic design rules** in the user interface above.

No alignment between field captions.

Heavy use of dividing lines, rather than whitespace.

Crowding.

Poor contrast between field captions and field contents.

No margins inside buttons (the "…" buttons and the Next Tab buttons).

No margin around Mailto/CGI radio button panel.

The "…" buttons on the right group with each other (Gestalt similarity) rather than with their fields.

5. Do a heuristic evaluation of the user interface above, finding at least 3 different problems (which must be different from problems listed in the previous question).

Mailto, CGI , URL are technical terms (match the real world, minor)

"max length = 200 characters" puts the counting burden on the user (error prevention, major)

Some captions end with colons, some do not (consistency, cosmetic)

Colors are represented as hex numbers, rather than color swatches (match the real world, major)

Alternate field is enabled even though it shouldn't apply (error prevention, major)

6. Suppose you are developing a stock trading application, in which delays of seconds can result in money lost. Thinking about the most important aspect of usability for this problem, discuss the applicability of each evaluation method listed below. Mention at least one limitation of each method that's relevant to the application you're developing.

(a) Heuristic evaluation

Heuristic evaluation can find some efficiency problems (such as missing shortcuts or obvious Fitts's Law problems), but it can't actually predict the efficiency of the interface. For example, the expert may not have enough domain expertise to know what the common operations are that need to be fast.

(b) Formative evaluation

Formative evaluation may not be good at finding efficiency problems, because the users studied are typically not experts, and think-aloud (typically used in these studies) tends to slow the user down.

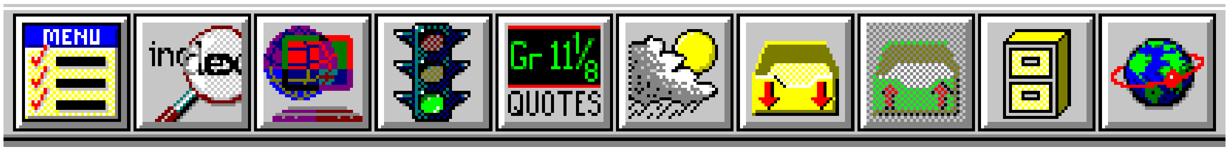
(c) Predictive evaluation

Predictive evaluation, such as the keystroke level model or CPM-GOMS, is designed for evaluating the efficiency of expert performance, which is what we want. Some models are less accurate than others, however; for example, the keystroke-level model doesn't consider parallelism, and both models assume error-free performance.

7. Suppose you're redesigning the appearance of an important part of your interface, so that the important part passes the squint test. Which visual variables will help, and which ones won't?

Selective variables will help pass the squint test: hue, value, position, size, orientation, texture.

Nonselective variables won't help: shape.



8. Discuss the usability of the color choices in the toolbar above.

Highly-saturated colors are tiring and distracting.

Too many colors make the buttons harder to scan.

Color-blind users might have trouble seeing the red-on-green arrows, or distinguishing the green light in the traffic light.

9. Discuss the internationalization of the toolbar above.

Menu, Index, QUOTES, and Gr would have to be translated to a local language.

The number "11 1/8" may need to be reformatted.

Inbox, outbox, and filing cabinets are everyday objects that may look different in other places.

Traffic light may also look different in other countries.

10. Suppose you have the MIT logo (shown below on the left) loaded into an Image object. The red and gray pixels are opaque, and the white background is transparent. You want this logo in all blue letters instead, but still with a transparent background (as shown on the right). Fill in the code below to make it happen. Some useful methods are shown at the bottom of the page.

Original logo

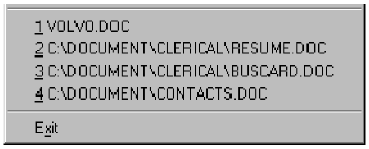
You want this

```
// Variables you can use
Image logo;     // the MIT logo
int w = logo.getWidth();
int h = logo.getHeight();

// Create a new Image for the result.
// The blue logo should be in the result image at the end.
Image result = new BufferedImage(w, h);
Graphics2D g = (Graphics2D) result.getGraphics();

// Your code follows
  g.setColor(new Color(0,0,1.0,1.0))
  g.fillRect(0,0,w,h)

  g.setComposite(DST_IN_SRC)
  g.drawImage(logo, 0, 0);


}
```

// Useful Swing methods
g.drawImage(*image, x, y*)
g.setColor(new Color(r, g, b, a))
where *r,g,b,a* are doubles in the range [0,1]
g.setComposite(*rule*)
where *rule* can be SRC_OVER_DST, DST_OVER_SRC,
SRC_IN_DST, DST_IN_SRC,
SRC_OUT_DST, DST_OUT_SRC
g.fillRect(*x,y,width,height*)

11. Make keystroke-level models of two methods for opening a file in a typical word processor. Show your work by annotating each step with the specific action the user takes. You don't have to estimate the actual time for each method. Assume the user has been typing prior to this method.

(a) Using **only the mouse** to choose volvo.doc from a Recent Files list at the bottom of the File menu (like the one below).


1 VOLVO.DOC
2 C:\DOCUMENT\CLERICAL\RESUME.DOC
3 C:\DOCUMENT\CLERICAL\BUSCARD.DOC
4 C:\DOCUMENT\CONTACTS.DOC
Exit

H (home to mouse)
M (mental operation)
P (point at File menu)
K (click to open File menu)
M
P (point at volvo.doc)
K (click on volvo.doc)

total: 1H + 2M + 2P + 2K

(b) Using **only the keyboard** to invoke the Open File dialog box using a shortcut (Ctrl-O or Command-O) and then typing volvo.doc explicitly.

M (mental)
K (Control or Command)
K (O)
M (mental)
9K (volvo.doc)
M (mental)
K (Enter)

total: 3M + 12K

12. Here is part of a Java program intended to animate a rectangle:

```
// instance variables
int x, y, w, h;  // current position of rectangle

// paint method
public void paint(Graphics g) {
    g.setColor(Color.RED);
    g.drawRect(x, y, w, h);
}

// animation method
public void animate(int newX, int newY) {
    int oldX = x, oldY = y;
    for (int i = 1; i <= 100; ++i) {
        Thread.sleep(10); // sleep for 10 msec
        x = (int) (oldX*(i/100.0) + newX*(1 - i/100.0));
        y = (int) (oldY*(i/100.0) + newY*(1 - i/100.0));
        paint(getGraphics());
    }
}
```

Suppose (x,y) is currently (0, 0). How will the program behave when it calls animate(100, 100)? Consider both **input** and **output** behavior, and draw pictures of the start, middle, and end of the animation to illustrate the output part of your answer.

Output: The program will leave the original rectangle at (0,0) without erasing it, and will display a smear of rectangles from (100,100) back to (0,0).

Input: no user input events will be handled for the duration of the animation (1 second), and then all the queued-up input events will be handled.

13. List three different things you should say to a user before a formative evaluation.

We're testing the interface, not you. Any problems you find are the system's fault, not yours.
You're free to stop at any time.
You are being recorded on videotape (or audiotape).
Your results will be confidential.
Please think aloud.

14. Louis Reasoner has 10 boolean preferences to display on a Preferences dialog box (preferences like Open Application Maximized and Save Automatically On Exit). He says, "Let's use a standard multiple-selection listbox containing all 10 choices." Give two reasons why his suggestion is a bad idea, and propose a better design.

Poor learnability: many users don't know the Ctrl-click convention for toggling choices on and off.

Poor error prevention: accidental clicking will clear all the preferences.

Inconsistent: use of a listbox suggests that the 10 preferences are related to each other, when they're not.

10 independent checkboxes would be a better design.

15. Consider this Javascript code that generates HTML:

```
  div.innerHTML = "Good " + (hourOfDay < 12 ? "morning" : (hourOfDay < 18 ?
"afternoon" : "evening")) + ", " + username + "!";
```

Aside from the fact that the string literals are English, what assumptions does this code make that might need internationalization?

Greetings are dependent on the time of the day.
There are 3 periods in a day, divided at 12:00am, 12:00pm, and 6:00pm.
The greetings are composed of a prefix and a variable suffix.
Punctuation: comma and exclamation point.
"Good" might have to change its form to agree with the noun (e.g. in gender).
Ordering of greeting and user's name might be different in different languages.

16. In a certain toolkit, undoable edits implement an interface that has 3 methods:

```
void doEdit();
void undo();
void redo();
```

Why are there both doEdit() and redo()? That is, should redo() not always be the same as doEdit()?

redo() needs to bring the user's attention to where and how the edit is being redone, possibly using animation.

Also, doEdit() might place the edit on the undo history, which is not something we want redo() to do.

END OF QUIZ

Extra space if needed

Extra space if needed