

Quiz 1

This quiz is closed book, closed notes. You have 80 minutes to complete it.

Your name: _____

1. (4 points) Suppose you're building an interface for air traffic controllers, who direct airplane traffic in flight. Which dimensions of usability are very important, and which are less important, and why?

Important: errors (people could die!), efficiency

Less important: memorability (controllers use it every day), learnability (lots of training), satisfaction

2. (3 points) What stages of the waterfall model typically involve users? What's wrong with this?

First stage (requirements) and last stage (acceptance testing or release). If you discover that you made a mistake in requirements or design, you've done a lot of work (written a lot of code) that you might have to throw away.

3. (4 points) What's the difference between the spiral model and just running a waterfall model iteratively?

The initial iterations of the spiral model use cheap prototypes and evaluation. Only later iterations use full-blown implementations, like the waterfall model does.

4. (3 points) List at least two cheap prototypes made in the development of the Olympic Message System.

scenarios

user manual

cardboard kiosk

Wizard of Oz prototype (with a human reading the system's lines)

5. (4 points) When you're observing users for task analysis, why is it particularly important to ask *why* the user is doing each task?

To extract the user's goal, because there may be a better design that achieves that goal than the procedure the user is currently following.

6. (4 points) In the MVC pattern, list the main responsibility of the model, view, and controller.

model: backend application state

view: output

controller: input

7. (4 points) Explain how the view hierarchy (also called component hierarchy) affects event dispatch and event propagation.

An event is dispatched to the deepest component in the hierarchy that contains the mouse pointer (or has mouse capture, or has the keyboard focus, in the case of key events). This dispatch target is found by a top-down scan over the component hierarchy. If two siblings both contain the mouse pointer, the sibling with higher z-order wins.

If the original target component chooses not to handle the event, the event propagates upward through the hierarchy – first to its parent, then the parent's parent, and so on, until either some component handles the event or it falls off the root, ignored.

8. (4 points) In the observer pattern, why would you want to make a *copy* of the list of observers every time a listener is added or removed?

In case the list of listeners is currently being iterated – which would be true if the add or remove was called in the context of an event handler for the same event. Most iterators don't allow the list to be modified while the iterator is in use.

9. (3 points) User interfaces don't need to have zero response time to appear instantaneous. What is the necessary response time, and what mental phenomenon is responsible?

About 100 ms, because of perceptual fusion.

10. (3 points) State Fitts's Law.

The time T to point at a target of size S at a distance D from the mouse (or finger) is given by $T=a+\log(D/S)$.

11. (3 points) What is the most common kind of color-blindness, and how common is it?

Red-green color-blindness, which occurs in 8% of males and 0.4% of females.

12. (4 points) Alyssa Hacker is trying to decide between using the component model and the stroke model for output handling. How does this decision affect input handling?

Components can have event listeners attached to them, and the component hierarchy automatically handles event dispatch and propagation.

Strokes require you to do the hit-testing (event dispatch) yourself.

13. Louis Reasoner implements his own scrollbar, which looks exactly like a normal scrollbar, but he neglects to use mouse capture when he's writing the mouse event handling.

A. (3 points) What will this mean for the user of the scrollbar?

If the user starts dragging the scrollbar thumb, but then the mouse pointer exits the scrollbar area, the thumb will stop moving in response to the mouse. The user must keep the mouse inside the scrollbar to drag the thumb.

B. (3 points) How much slower will it be to use Louis's scrollbar than a scrollbar implemented properly? (Give an analytical relationship, not an exact number.) Explain.

*Without mouse capture, moving the scrollbar thumb to a particular spot becomes a **steering task**, where the user must keep their mouse constrained to a tunnel. So it's governed by the linear relationship $T=a+b(D/S)$, where S in this case is the width of the scrollbar.*

With mouse capture, moving the thumb is a Fitts's Law task, so it's governed by a logarithmic relationship $T=a+b\log(D/S)$. (The task still involves error correction if the user is trying to position the thumb to a particular place, but it's not tiny cycles of error correction, just one big cycle.)

So there's an exponential slowdown, at least in terms of D , the distance scrolled.

C. (3 points) Ben Bitdiddle says, "If you always put Louis's scrollbar at the edge of the screen, it won't be any slower." What does he mean?

Because the edge of the screen stops the mouse pointer but not the user's hand, it makes the steering tunnel infinite width, changing the task back to an unconstrained pointing task.

14. (3 points) Which of the following events are raw input events (not translated events)?

- mouse pressed
- mouse released
- mouse clicked
- mouse moved
- mouse dragged

15. Louis Reasoner is writing an interface that draws checkers on a checkerboard. He's decided to represent the board itself as a component, a checkerboard square as a child component of the board, and a checker as a child component of a square.

A. (3 points) Louis notices that when he drags a checker outside its square, it disappears. Explain why.

The checker's output is clipped by its parent square's bounding box.

B. (3 points) Louis claims, "My representation should allow minimal repainting when a checker moves by itself" (i.e., using board.move(), not mouse dragging). Assuming he's right, and his design only paints the minimum number of components necessary, which components will have their paintComponent() methods called when a checker moves by itself?

Four components total: the checker itself, the origin and destination squares, and the board.

C. (3 points) Ben Bitdiddle argues, "But if Java only uses a single damage rectangle for the entire window, it will always repaint more than that when a checker moves diagonally." Explain why.

The damage rectangle for a diagonal move will include at least two other squares (along the opposite diagonal), so those squares and any checkers in them will also be repainted.

16. (3 points) Which dimensions of usability does a metaphor help, and which dimensions are irrelevant?

*Helps learnability and memorability.
Irrelevant to errors, efficiency, and satisfaction.*

17. (3 points) Louis Reasoner says, “A command language should have no more than about 7 commands.” Where did he get this number, and why is it irrelevant to what he’s talking about?

7 chunks is the average size of working memory. It’s irrelevant because you don’t have to store the entire command language in working memory; either you store it in the world (in manuals or quick reference sheets) or in your long-term memory.

18. (4 points) List all the ways that a speech recognition interface over a telephone *fails* to meet the definition of direct manipulation.

*No continuous visual representation;
No physical actions to manipulate the representation;
No rapid, incremental, or easily-reversible effects*

19. (3 points) Give two ways that the conventional scrollbar demonstrates natural mapping.

*Up and down arrow buttons are above and below the scrollbar track, respectively.
Moving the thumb up and down moves the scroll view up and down, respectively.
Proportion of thumb to track equals proportion of visible view to entire document.
Clicking in track above the thumb pages up, and below the thumb pages down.*

20. (3 points) Define *mode error*.

The user performs an action that has an unexpected effect because the system is in a state (mode) where the action means something different than the user expected.

21. (3 points) Ben Bitdiddle says, “Improvements to widely-used user interface techniques should be invisible, like Apple’s improvement to the way cascading submenus work.” Which Nielsen heuristic justifies his statement?

Consistency. (Credit also given for minimalist design.)

22. (3 points) Alyssa Hacker says, “Invisible modes are bad.” Which two Nielsen heuristics justify her statement?

Error prevention and visibility of system status.

23. (4 points) Suppose a new web site is prototyped by drawing up pages in an HTML editor, one for each important screen, and printing them out to show to users. State the fidelity of this prototype for each of the four dimensions of fidelity.

High breadth, low depth, high look, low feel.

24. (4 points) List one important danger of paper prototyping, and one important danger of computer prototyping.

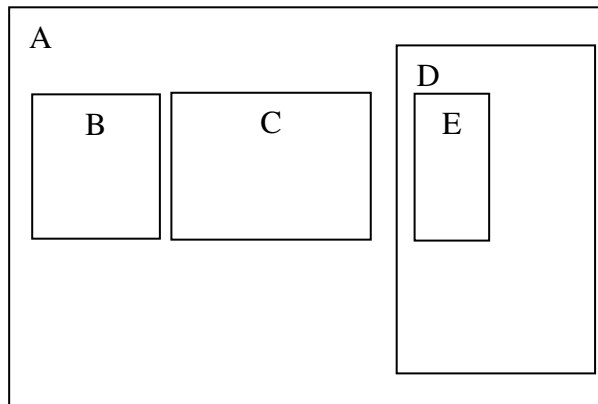
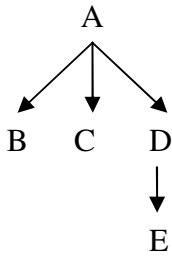
A paper prototype may require a backend that’s infeasible to implement.

A computer prototype may contain sloppy code that’s never thrown away, but instead reused in the final implementation.

25. (3 points) Define *participatory design*.

One or more target users are actual members of the design team, contributing design ideas, not acting as test subjects.

26. (3 points) In the layout shown below (view hierarchy on left, screen appearance on the right), suppose B, C, and E should always have equal heights. Why is this requirement impossible to enforce with layout managers?



Layout managers can only enforce constraints among siblings (and their parent) in the view hierarchy, but B, C, and E do not share the same parent.

END OF QUIZ