Quiz 1

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: _____

1. (6 points) Here are three user interface design problems. Which dimensions of usability are most important for each problem, and why?

   a) A nuclear power plant's control console.

   ***Errors**, because mistakes can cause radition accidents and loss of life. Also **efficiency**, because emergency situations may need to be understood and handled quickly. Learnability, memorability, and subjective satisfaction are less important, because users are trained, frequent users of the interface.*

   b) The flight controls of a passenger airliner.

   ***Efficiency**, because flying is a highly-skilled, high-performance task; also **errors**, because mistakes cause crashes and loss of life. Again, learnability memorability, and satisfaction are less important.*

   c) A web site for making a Reg Day appointment with your advisor.

   ***Memorability,** because the task must be done repeatedly but with long time periods between. **Learnability** would also be reasonable, because a highly learnable interface doesn't need to be remembered, and users won't be trained in it. Errors are less important because their impact is low, and efficiency and satisfaction are least important.*

2. (3 points) How is perceptual fusion related to computer response time?

   A computer response within the perceptual fusion interval (~ 100 ms) will be perceived as instantaneous, so no additional feedback is required.

3. (6 points) Suppose you are designing a kiosk for people taking the written part of their driver's license test. Do a user analysis for this problem below. Be as thorough as you can.

User classes:

Teenagers *getting their license for the first time. Age 16-18, more familiar with computers (and test taking) than other user classes.*

Immigrants *moving into the country. Age 18-60 or so, may have limited or no English skills.*

Adults *who never learned to drive or bothered to get licenses when they were younger. Age 18-60 or so.*

Elderly *drivers being retested because of their age. Age 60+. May have reduced eyesight, limited computer skills.*

Clerks *at the motor vehicle bureau helping other users use the kiosks and dealing with system problems. Age 21-40 or so. Can be trained with interface.*

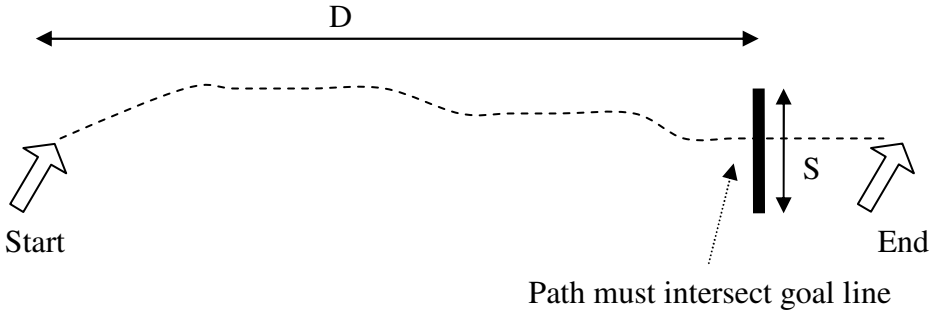
Acceptable answers have both age-based classes and role-based classes, with specific characteristics and insights about each class.

4. (3 points) On his first day on the job, Joe Hacker is handed a detailed specification of a user interface to implement. If his company is following the waterfall model, then where did this specification come from, and what's the risk? If his company is following the other model we discussed in class, then where did this specification come from?

   With the waterfall model, the specification comes from requirements analysis (which involves users) followed by a design stage (which doesn't involve users).

   With the spiral model (or iterative design model) discussed in class, the specification should have come from one or more iterations of cheap prototyping (such as design sketches or paper prototyping) and evaluation (such as user testing).

5. (3 points) Suppose that a user interface requires the user to move their mouse or pen so that its path intersects the goal line, as shown below. This is called a *goal crossing* task, in analogy to crossing a soccer goal line. What is the index of difficulty of this task as a function of D and S? (Constants can be omitted.)



   ID = log(D/S). This is analogous to a Fitts's Law pointing task; see the Accot & Zhai reading.

6. (3 points) The *lasso selection* tool in a graphical editor allows the user to trace the outline of a shape to select. How does the time *T* that it takes to make a lasso selection depend on the perimeter *P* of the object (measured in pixels) and the desired precision *C* (also measured in pixels)? Explain how you found your answer.

   T is proportional to P/C. This is a steering task, where the user must keep the pointer constrained to a tunnel of width C whose length is the perimeter P.

7. (3 points) Dale says, "Never put red and blue next to each other in an interface, because it may be hard for some people to tell them apart." Do you agree with this statement? Explain.

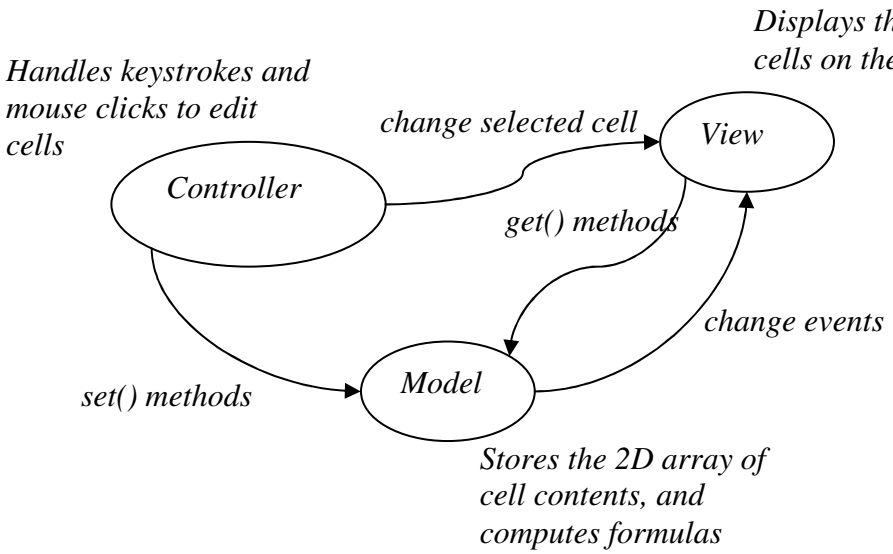
   It's not good to put red and blue next to each other, but not for the reason Dale gives. Red and blue exhibit chromatic aberration, which makes the edge between them hard to focus and causes eyestrain from changing focus. Red and ***green*** *are the colors that most color-blind people can't distinguish.*

8. (3 points) You're studying an implementation of an MVC design, and you notice that the model is calling repaint(). What's wrong with that? When would it be okay for a *controller* to call repaint()?

   The view is responsible for its output, so it should call repaint(), not the model. Another way to put it is that this hardwires a particular view to the model.

   It's OK for a controller to call repaint() if the controller is tightly coupled to the view, e.g. if it needs to draw feedback or affordances on the view.

9. (3 points) Draw a model-view-controller architecture for a spreadsheet, with a brief description of the specific responsibilities of each part.



An architecture that fused the controller and view into a single object would also be reasonable.

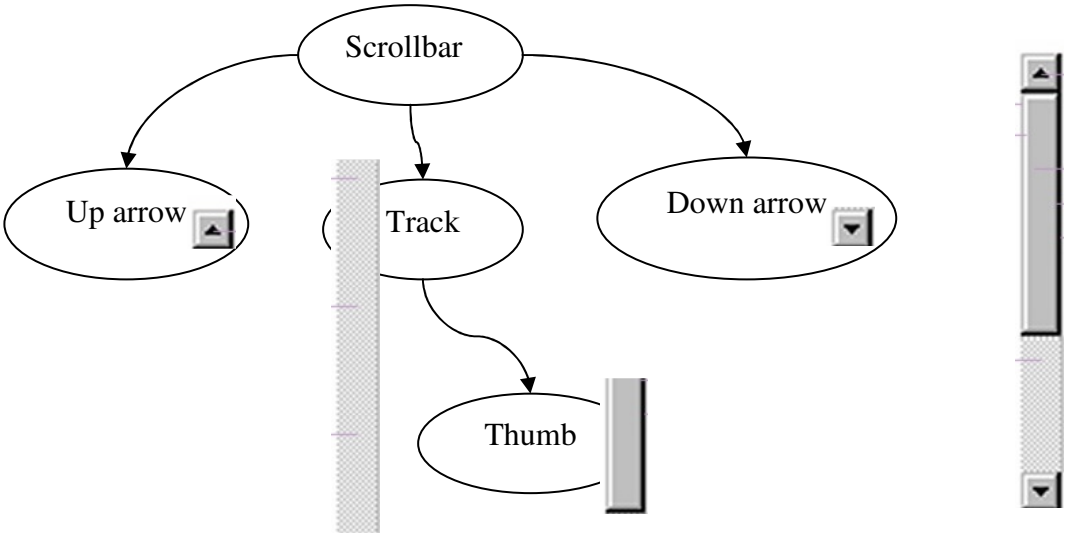
10. (3 points) Describe 2 important issues in the design of the interface between the spreadsheet's model and view.

(1) The view should be decoupled from the model, so that changes in the model are sent to the view by events rather than direct method calls on the view. (2) Events might be fine-grained (cell-by-cell) or coarse-grained (entire spreadsheet) or something in between (rectangular areas). (3) Listeners to the events might likewise register interest at a fine-grained level (only events about this cell) or coarse-grained (events about any cell in the spreadsheet). (4) The get() methods that access the model can likewise be fine (returning one cell) or coarse (returning an array of cells all at once).

11. (3 points) Louis Reasoner proposes skipping the event interface between the spreadsheet's model and view, and simply having the controller tell the view to update itself whenever the user edits a spreadsheet cell. What's the flaw in his idea?

Controller only knows about changes to cells that the user edited directly. If changes happen in other cells as a result (e.g. because of formulas), only the model knows about those changes.

12. (3 points) Suppose the scrollbar shown on the right is internally implemented by at least 4 components. Draw a view hierarchy for it, labeling each object with its function and a little picture of what it should paint on the screen. The top-level component is shown.



Other arrangements are possible, but order matters: the thumb must be on top of the track, so it must be either a child or later sibling of the track.

13. (3 points) In order to produce the usual scrollbar behavior, which components would you attach mouse event handlers to, and (briefly) what would each handler do?

Up arrow: on mouse click, scroll up one line. (Best answer: on mouse press, scroll up one line, and set timer that autorepeats until mouse release.)

Down arrow: same, but scroll down instead of up.

Thumb: on mouse drag (press and move), scroll the view to match the thumb's position relative to the track. (Best answer: use mouse capture to keep getting move events even if mouse strays from the thumb.)

Track: on mouse click, scroll up by one page if mouse is above thumb, down by one page if mouse is below thumb. (Best answer: use autorepeat as for arrow buttons.)

14. (3 points) Explain how z-order affects drawing and input handling.

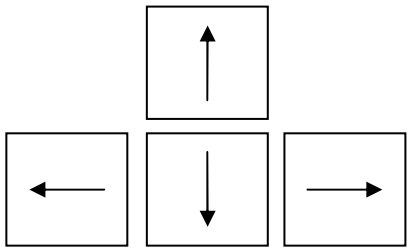
Z-order is the layering order of components in the view hierarchy. For drawing, parents are drawn underneath children, and earlier siblings in the z-order are drawn underneath later siblings. For input handling, the component with the highest z-order (i.e. on top) at the mouse position receives the initial event dispatch, but then event propagation occurs through ancestors, ***not*** *necessarily down the z-order.*

15. (3 points) Louis Reasoner is writing the paint() method for his application window, and for fun he decides to fill the entire screen with a shocking green color. He sets the color correctly, determines the width and height of the screen correctly, and then makes a call to fillRect(0,0,screenWidth, screenHeight). But fillRect() doesn't do what he expects it to. Give two reasons why he's wrong.

(1) The fillRect() call doesn't fill the whole screen, because his drawing is clipped to his window. (2) The coordinate system is with respect to the window, so (0,0) is the upper-left corner of the window, not of the entire screen.

(There are many other things that Louis ***might*** *have done wrong, of course, including forgetting a semicolon or a curly brace so that his program wouldn't even compile. Only answers about what he* ***definitely*** *did wrong receive full credit.)*

16. (3 points) Consider the conventional "inverted-T" arrangement of keyboard arrows shown below. Is it internally consistent? Is it metaphorically consistent? Explain.



No, it isn't internally consistent, because the up/down arrows are next to each other while the left/right arrows are separated. (Also, the numeric keypad arrows are arranged differently.) One can also argue that it's internally consistent in that each button is labeled with an arrow correctly identifying its function.

And no, it isn't metaphorically consistent either, because similar directional designs in the real world (like a compass rose) do not pack the arrows tightly together like this. This example was used in the Grudin reading.

17. (4 points) The code below shows a simple model that fires events. Find two bugs in the event-firing mechanism, and explain why.

```
public class Square {
  private double side;
  private double area;
  private Set<SquareListener> listeners = new HashSet<SquareListener>();

  public Square(double width, double height) {
    this.side = side;
    this.area = side * side;
  }

  public double getSide() { return side; }
  public double getArea() { return area; }

  public void setSide(double side) {
    this.side = side;
    for (SquareListener l : listeners) { l.squareChanged(this);  }
    area = side * side;
  }
  public void addListener(SquareListener l) { listeners.add(l); }
  public void removeListener(SquareListener l) { listeners.remove(l); }
}

public interface SquareListener {
  public void squareChanged(Square square);
}
```

*setSide() fires its events before satisfying its invariant (that area == side*side, so if the event handler calls getArea() it will get an incorrect value.*

Event firing iterates over the same collection mutated by adding and removing, so if an event handler adds or removes a listener, the iteration may fail. (In Java, this would cause a concurrent modification exception.)

There's also a bug in the constructor of Square. This isn't a problem in the ***event-firing mechanism****, but was accepted as an answer anyway.*

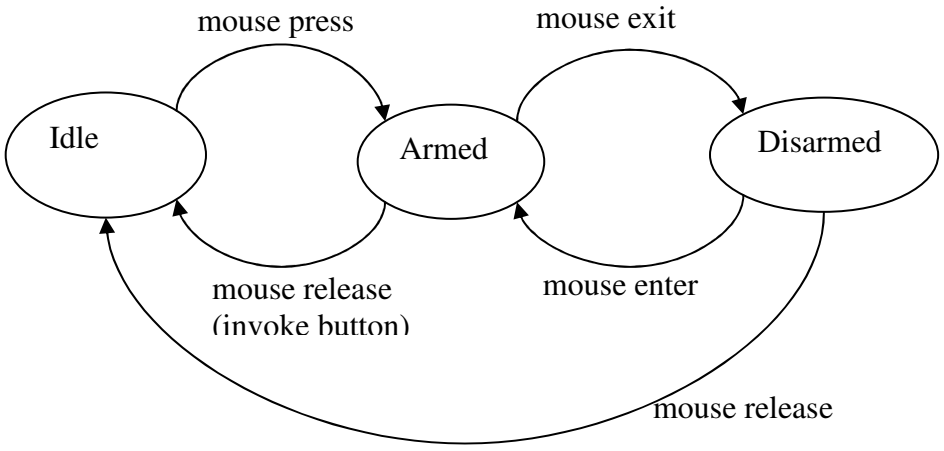
18. (3 points) Joe says, "When a web browser window is maximized so that its scrollbar is on the right edge of the screen, mouse capture makes it easier to hit it." Do you agree with this statement? Explain.

No. The scrollbar might indeed be easier to hit (as long as there isn't a window border separating it from the edge of the screen), but because the edge of the screen is an infinite-size Fitts's Law target, not because of mouse capture. Mouse capture makes it easier to ***drag*** *the scrollbar, by allowing the mouse to leave the scrollbar while dragging, so that instead of being a linear steering task, scrolling is a logarithmic pointing task.*

19. (3 points) Why are timestamps stored in events?

Timestamps are needed for handling or translating some events (such as double-clicks or keyboard autorepeat); the timestamp must be stored because events may not be handled in real time; they may sit in the queue for a time until the application handles them.

20. (3 points) Draw the state diagram for a command button, such as the OK button of a dialog box. Consider only mouse events; ignore keyboard events.



21. (3 points) Briefly state the responsibilities of the 3 roles for the design team in a paper prototype test session.

Computer: runs the prototype, responds mechanically.
Facilitator: interacts with the user, providing tasks and moving the session along.
Observer: watches quietly taking notes.

22. (3 points) List three dimensions of usability, and for each one, list one design principle that helps improve that dimension.

Learnability: match the real world, consistency, minimalist design.
Efficiency: flexibility & efficiency, Fitts's Law
Memorability: recognition not recall
Errors: error prevention, recognition not recall, error reporting
Satisfaction: aesthetic & minimalist designx

Other design principle answers are possible.

23. (3 points) Describe a mode found in every automobile interface that could lead to a serious mode error.

The gearshift state, specifically forward gear vs reverse gear. Pressing the accelerator causes the car to go opposite directions in each mode. If the driver is in the wrong mode, the car might collide with something.

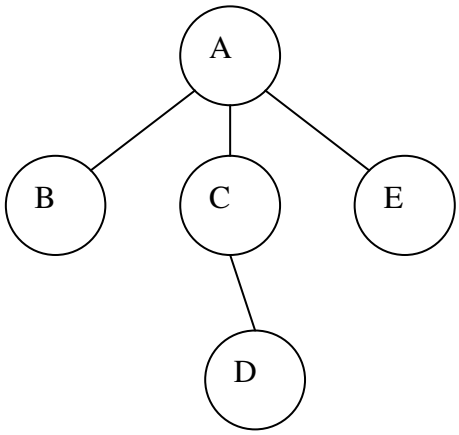
24. (3 points) Is a web page a direct manipulation interface? Explain.

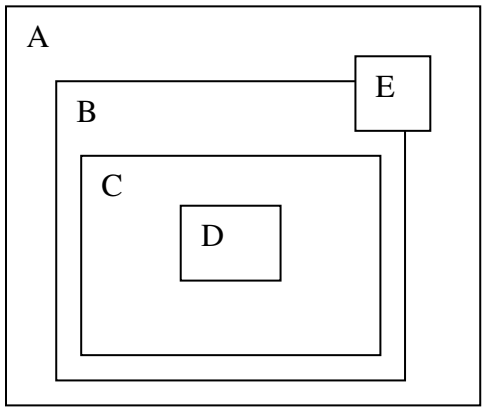
Yes and no. A web page has a continuous visual representation, which the user controls by labeled button presses (hyperlinks). However, responses are not always rapid (the network can be slow) or incremental (the entire page usually changes), but usually are reversible (by the Back button).

25. (3 points) What are the primary affordance cues that tell a user how to use a web page?

Hyperlinks (blue underlined text) are the primary cues. The mouse cursor also changes when it hovers over a hyperlink.

26. (6 points) Suppose the component hierarchy shown below is implemented in a conventional GUI toolkit, like Java Swing.





a) If the entire area of C is damaged, which components need to be repainted, and in what order?

A, B, C, D. (E doesn't need to be repainted because it doesn't intersect the damage region.)

b) If the entire area of E is damaged, which components need to be repainted, and in what order?

A, B, E. (C & D don't intersect the damage region.)

c) If D has the keyboard focus, and the user clicks on C, then which components can receive the click event, and in what order?

C, A. (Only ancestors of C in the component hierarchy can receive the click event.)

27. (3 points) Suppose you are implementing the output view of a text editor. How could each of the three output models we discussed be used to implement text display?

Component model: each character might be a component, as in the glyphs reading.
Stroke model: each character is represented by a set of drawn curves.
Pixel model: each character is a pixel image.

28. (3 points) In both the RGB and HSV color models, (0,0,0) is **black**. What is **white** in the RGB and HSV models?

RGB: (100%,100%,100%)
HSV: (anything,0%,100%).

29. (3 points) You want to make a paper prototype of the interface you're designing, but your colleague doesn't. Give two reasons to support your position, and one reason that your colleague might give against it.

In favor of paper prototype: fast & cheap to build; easy to change on the fly; focuses attention on questions important in early design, must be thrown away; nonprogrammers can help; can be both deep and broad.

Against: paper prototypes can't test many things (e.g. interactive feedback, whether small changes are noticed, Fitts's Law questions); user may work too deliberately.

30. (3 points) What's similar about a paper prototype and a Wizard of Oz prototype? What's different?

Similar: both have a human backend.

Different: frontend of paper prototype is on paper; frontend of Wizard of Oz is higher fidelity computer interface..

END OF QUIZ