

Sample Midterm

[This sample exam is representative of the actual midterm, but the type, form, topics, and number of problems on the actual midterm may not be the same.]

Please read all the problems carefully. Do everything we ask for, neither more nor less. If you have any questions on what a problem means, don't hesitate to ask. Don't get bogged down on any one problem; there are 54 points possible, so allowing approximately one minute per point will give you some time left over. If you have trouble on a problem, go on to the next one. When you're done, you may turn in your exam and leave.

In answering these questions, you may use any paper materials that you brought with you. You may not share any materials with classmates during the exam and you may not use any electronic devices.

Use the correct HCI terminology wherever possible in your answers and describe how those terms and principles apply to the specific facts stated in the problem.

Please write your answers clearly and neatly—we can't give you credit if we can't decipher what you've written. Use the backs of the pages (or any other paper) for scratch work, but cross out any work that you don't want us to consider as part of your answer (and if your answer doesn't appear entirely within the indicated answer space for a question, indicate clearly where the rest of the answer is). We'll give partial credit for partially correct answers, so writing something is better than writing nothing. But we will also deduct for irrelevant, extraneous information, even if the correct answer is buried somewhere within it, so don't just write down everything you know. We have given length guidelines for many answers; ignore them at your own risk (but aim for clarity and conciseness more than a specific number of sentences in your answer). Unless your handwriting is huge, the space we provide for an answer should be more than enough. It is generally best to think more and write less.

Good luck!

Problem 1 (3 points) **Nielsen's principles**

Many applications use keyboard shortcuts (like control-S or command-S to save a file, or like using the tab key to move from one field of a form to the next).

(a) Keyboard shortcuts are an example of one of Jakob Nielsen's usability principles. Which one?

Flexibility and Efficiency of Use

(b) There's also one of Nielsen's principles that keyboard shortcuts *don't* follow. Which principle would favor menu items, for example, over keyboard shortcuts?

Recognition over Recall: You have to remember the keyboard shortcut.

(c) A music-creation web site that allows the user to specify notes by clicking on an image of a piano keyboard illustrates which of Nielsen's principles?

Match between the system and the real world.

Problem 1
(3 points)

Problem 2
(8 points)

Problem 3
(3 points)

Problem 4
(10 points)

Problem 5
(3 points)

Problem 6
(6 points)

Problem 7
(4 points)

Problem 8
(5 points)

Problem 9
(12 points)

Total
(54 points)

Problem 2 (8 points) **HCI epistemology: How we know what we know about HCI**

These require only very brief answers. **SCORING: 2 points each.**

(a) Why can't we expect to get good interaction designs by just relying on the HCI designer or developer's expertise and common sense?

Their expertise is in design and development; their common sense is based on their own experience. But good interaction design requires knowing what the USER knows, which often differs from the designer/developer's experience. (Also, some things are counterintuitive.)

(b) Why can't we use a "theory of HCI" that lets us look at an interface and tell how usable it will be (the way we can look at code through the "theory of algorithms" and tell how long it will take to run)?

Because there isn't one. We don't know enough about people, cognition, usability, at least not enough to generalize.

(c) So what is the primary technique/practice/principle we use to achieve good interfaces?

User testing; user-centered design; evaluation and re-evaluation, iteratively. The point is that we have to test things out on our users.

(d) What is the drawback/disadvantage of the approach in part (c)? In other words, why isn't it effective enough to give us usable interfaces every time?

Because testing only proves the presence of bugs (of UI flaws), never their absence; we don't KNOW the tested UI is problem-free. It would be NICE to have a theory.

Problem 3 (3 points) **Colors and Vision**

Alex and Sam are designing a web site for architects. Part of the site will give detailed building design plans (blueprints). Sam wants the blueprint page to use different shades of blue, but Alex wants to use black and white, with red where additional color is needed. Who has the better design, and why?

Alex is right; Sam is wrong. Blue is no good for fine detail because there are few blue receptors (cones) in the fovea, where detail work happens.

Problem 4 (10 points) **Pre-attentive and memory tasks/coding**

(a) (3 points) For each of the user tasks below, indicate whether it's an example of long-term memory, short-term memory, or pre-attentive processing. **SCORING: 1/2 point each**

(a.1) The status panel in a fantasy game flashes yellow when a monster is nearby. **PreAtt**

(a.2) An instructor tells the class a special Email address for course-related questions **Long-term**

(a.3) An on-line course registration system displays a five-digit course number; to enroll, the user has to type that number on the next screen **Short-term**

(a.4) A traveler buys one-day wi-fi access at the airport and receives a login ID and password. Every 15 minutes or so, the system disconnects the traveler, who then has to log in again **Long-term**

(a.5) An industrial robot's status display's background turns red when the robot is low on oil **PreAtt**

(a.6) An on-line restaurant menu lists appetizers, main dishes, drinks, and desserts, each in a separate column **Pre-attentive**

(b) (2 points) Pick any two of the memory-related (non-pre-attentive) items above and, for each, describe a change to the system that would reduce or eliminate the memory burden.

Email address: Put it on syllabus

Reg system: Allow copy/paste or combine onto one screen (so user can check by the number, e.g.)

Wi-fi: Save the password locally on the machine (or make the connections more reliable)

(c) (3 points) We can use a variety of coding methods to represent different items visually.

(c.1) Which method—different colors or different three-letter strings—allows the user to distinguish a greater number of different items?

strings

(c.2) What is a possible disadvantage of using three-letter strings?

Hard to memorize, no good for people who don't know the Roman alphabet

(c.3) What is a possible disadvantage of using colors?

People could be color-blind; display could have low color fidelity.

(d) (2 points) Miller and memory limits

Which of the following is justified by George Miller's rule of 7 ± 2 ? Circle the one best answer.

- A. Don't design PowerPoint slides with more than 9 bullet points; instead, split the points over multiple slides.
- B. Don't have fewer than 5 items in a menu.
- C. Don't expect a user to remember a random 10-digit password from one screen to the next. **THIS ONE**
- D. Don't design web pages with dozens of links.
- E. Don't require users to memorize 10-digit telephone numbers.

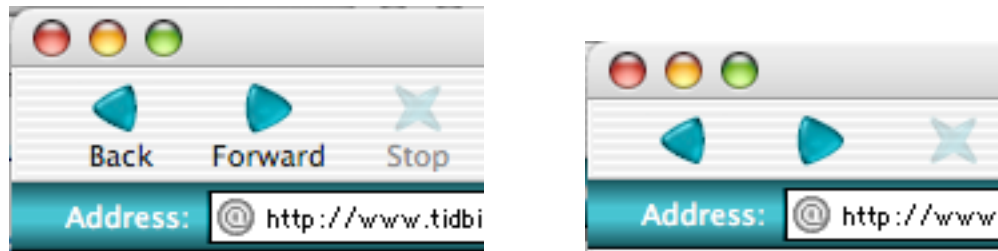
Problem 5 (3 points) Nat. Lang [SCORING: -1 for each line incorrectly checked or incorrectly unchecked. Min score 0.]

Some of the following are reasons why natural language understanding is still a difficult problem for computer systems. Some are not. Check each item that *does* help explain why natural language understanding is hard.

- ☐ Natural language is ambiguous, and a lot of real-world context is required to disambiguate sentences.
- ☐ Continuous speech is not easily divisible into discrete individual sounds or words.
- ☐ Natural language is rich in its ability to paraphrase—there are many possible variations for expressing the same meaning.
- ☐ Most modern programming languages have poorly developed facilities for string handling. **NO**
- ☐ Individual speakers' voices vary greatly due to age, gender, native language, and regional dialect.
- ☐ Past attempts at natural language understanding have been so unsuccessful that hardly any researchers work on the problem these days. **NO**

Problem 6 (6 points) *[This question inspired by Bruce Tognazzini]* **Fitts's Law (Bruce Tognazzini)**

(a) (2 points) These two screen shots show a web browser's toolbar with a preference set in two different ways:



Which setting—with or without the text labels—provides faster access to the back/forward/stop tools? Give one sentence justifying your answer.

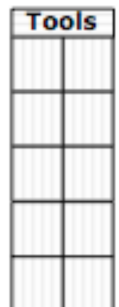
With text labels is faster because the target is bigger. They could also say that there's faster access because the text is more readily recognizable (that's not Fitts' law, but it's still true).

(b) (2 points) Which provides faster access, one menu bar at the top of the screen or a menu bar at the top of each window? Give one sentence justifying your answer.

Top of the screen: According to Fitts's Law, access speed is a function of the distance to the object and its size. The size is an issue because users have to slow down to hit small things. The size of items at the top of the screen is effectively huge, because there's no danger of overshooting.

(c) (2 points) If you have a toolbar at the right edge of your screen (like the one shown here), containing ten 16x16-pixel icons in two columns of five, how could you rearrange these icons for faster average access? Again, explain your answer.

Spread them out along the right edge in one column of ten; then you get the edge effect for all.

**Problem 7** (4 points) **Prototyping**

(a) (2 points) You want to evaluate alternative web page designs. The issues you want to focus on are choice of typeface, color, and overall readability of the site. Would you use a high-fidelity prototype or a low-fidelity prototype, and why?

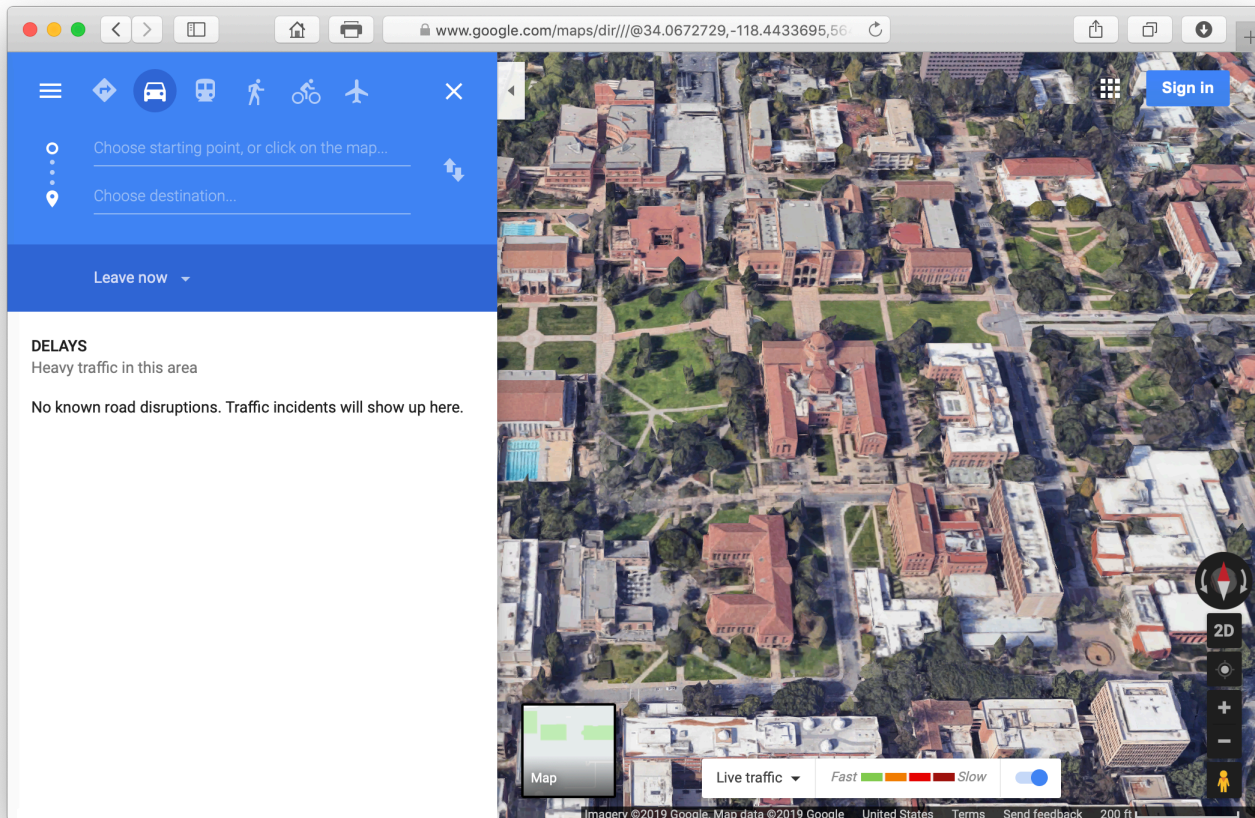
High-fidelity: The issues you're interested in are all issues of final form.

(b) (2 points) You want to evaluate alternative web page designs. The issues you want to focus on are the organization of the site and the overall ease of navigation. Would you use a high-fidelity prototype or a low-fidelity prototype, and why?

Low-fidelity: You don't want the user to focus on the "finish", you don't have time or money to build out both alternatives fully, if you want to redesign as a result of your evaluation, you don't want to have to throw out much investment.

Problem 8 (5 points) **Affordances**

Below is a screen shot from Google Earth.



(a) (3 points) Identify three of the best, clearest affordances on this image for users of this application (don't choose two instances of the same kind of control). For each, say what action or function it affords.

Buttons for clicking ("Sign In" and black ones in lower right corner). Text entry area for typing (underscores with text for start and destination). Down-pointing triangles for expanding /pull-down menus ("Leave now", "Live traffic") "X" for clicking to close directions panel. Up-and-down arrows to swap start and destination. Compass to rotate to change angle of view. [SCORING: 1/2 point for each item, 1/2 point for the item's function. Answer must be in terms of affordance---i.e., what action/function it makes obvious.]

(b) (2 points) Identify one control on this page that could be a clearer example of an affordance, and say what could be clearer about it. (This doesn't have to be something that's terribly confusing; it just has to be capable of improvement as an affordance.)

Current location circle. Slider/toggle in Live Traffic bar at bottom. Could be a lot of things. [SCORING: Again, not any UI improvement, but one that makes the item's function more obvious.]

(c) (0 points, just for fun) Draw an X on the map above to indicate where you are right now.

Problem 9 (12 points) **Requirements**

Different applications have different HCI requirements. Fill in the empty cells of the table below with brief descriptions of the various requirements for each listed task in each category. The cells are small, so you can't be complete; just list the characteristics that most distinguish each task.

Score 1 pt for giving a major distinguish'g characteristic. Score 1/2 pt if ans. plausibly fits category

Task: Category:	Web-based store	Fighter jet	Automatic flight check-in kiosk at airport	Smart glasses for people with dementia
Functional	Display merchandise, accept orders, interact with the rest of the order system	Respond to pilot's commands, fly safely and effectively, fire weapons and evade enemy fire	User enters ticket number, baggage info; kiosk may allow choosing seats; prints boarding pass	Recognize faces of nearby people; give directions to help user get around; detect hazards
Data	Merchandise in stock, customers' orders, possibly customer info and preferences	Characteristics of friendly and enemy aircraft and weapons, pilot preferences	Flights, passengers, available seats, baggage limits	Contact list with photos; GPS/map information (home, common locations), public transit schedule.
Environmental (physical, social, organizational)	Most anywhere, any type device. Usually used solo; no formal training, documentation	Embedded in aircraft, special controls, high noise and motion, very high training and support	In airport ticketing area; little or no support; typically very crowded; cluttered with baggage	Wearable (they're glasses), visible and audible with background noise outside, not distracting to other people the user interacts with
User	General public, teens/adults; familiar with web browsing but not nec. other tech.	Military pilot: highly trained, cohesive group, very frequent use	General public: little sophistication; perhaps infrequent use; anxious, hurried, stressed	Likely to be older, may have vision/hearing/mobility problems, may not be familiar with technology or common UI affordances, may be slow to react
Usability	Commerce requires trust, clear feedback, low frustration/high satisfaction	Immediate responsiveness, support split-second decisions, error prevention	Trivial learnability, visibility of status/clarity, high reliability.	Consistently reliable; trustworthy; easy to learn; efficient/simple (no complex command sequences); to help user avoid risky situations may need to take a cautious approach.