CS6750 Assignment P2

Lucas Rosen
lrosen7@gatech.edu

1 TASKS IN AN HOUR

1.1 Task list

Table 1—Task list for an hour on a train.

Task	Goals	Interfaces	Object
Sitting on the train	Getting to my apartment in Baltimore from New York	Seat, ticket, bag	Myself (I am moved from one location to another)
Listening to music	Pass time while I'm on the train	Spotify app, headphones, cell phone	Music that is playing on my phone
Reading for CS6750	Study for the class to do well on exams and get a good grade	Laptop, web browser, PDF reader	Text of the readings
Texting my brother	Send my brother a text about something I saw related to his job	Cell phone, text message app	The information pertinent to my brother
Showing the ticket to the train conductor	Show proof that I paid for the train	Cell phone, email app	Receipt of purchase of train ticket

1.2 Directness and invisibility

1.2.1 Listening to music

I used the Spotify app on my smartphone to listen to music. The interaction was very direct. After searching for a song, I clicked the title of the song I wanted to hear on my phone's touch screen.

I spent almost no time at all thinking about the interface. This is through good design. An example of this good design is the search feature. Being able to search for songs, artists, and playlists makes it easy to select what I want to hear. Spotify also has playlists automatically generated for me based on my taste. This removes the cognitive burden of deciding what I want to listen to. These two features together with the direct touch controls makes the task of listening to music invisible.

1.2.2 Showing my ticket to the conductor

I took a train from New York to Baltimore in which I had to show a ticket, proof that I paid to be on the train. In this case, the ticket was virtual – a QR code in an attachment from an email I received from the train company when I paid.

The interaction was very direct and required almost no manipulation of the interface. However, the interface was not invisible, and far less invisible than in the case of using a physical ticket, which can simply be pulled out of my pocket and given to the conductor.

For the virtual ticket, I needed to worry about having an internet connection to get to my email, and my phone having enough battery to display the ticket. I needed to worry about remembering how to pull up the email with the code. I also had to make sure that my screen was bright enough for the ticket to scan. Given that the actual task of showing the ticket took 2-3 seconds, I spent far more time thinking about the interface than the task.

1.2.3 Reading for CS6750

While on the train, I spent time doing readings for this class on my laptop.

The interaction was direct. I opened up the app, and then it read like a book.

I spent almost no time thinking about the interface and this is because of good design. The main actions that I need to take when reading (aside from reading itself) are zooming in/out and scrolling through the pages. Before laptops had gestures and touch pads, these were not easy. These design innovations (gestures and touch pads) made them easy.

Before pinch to zoom, I used *ctrl* + or *ctrl* – to zoom in and out. The issue with this approach is that the zoom changes are quantized (also, the specific button presses must be learned). One *ctrl* + press may zoom in too much, but then zooming out once goes back to the original state which was also not good. I would have to find a menu where I could manually type in a zoom percentage. This often requires trial and error and is very indirect. Pinch to zoom fixed these issues.

1.2.4 Texting my brother

Interaction via texting is indirect. Going from the idea in my head to getting that information in my brother's head is entirely mediated via the typing and reading of text on our cell phones.

Despite this indirectness, I spent very little time thinking about the interface and this is through both good design and learning. I have always found typing on a cell phone awkward and have had to think a lot about the interface. What has helped is auto-suggest where upon typing a word I get three suggestions of what I *really meant*. I can get most letters wrong in a word but if they were wrong in ways that make sense (I typed letters physically close to the right ones, for example) then the correct word is suggested. I can quickly type something close to the right word and the one I really want will very likely be suggested.

My heavy reliance on auto-suggest took some learning, however. I had to understand what kind of mistakes would work and what I needed to correct in real time because auto-suggest wouldn't pick it up. This learning took time (and is almost entirely procedural, even trying to write down the rules that I unconsciously follow is very tough), but is now second nature.

2 INVISIBLE TASK

The task I have selected is that of choosing a matching outfit.

2.1 Components I used to think about

The rules of what patterns matched with others and what colors matched did not come naturally to me. I would put on striped pants and checkered shirts. I mixed colors like blue and red which are both too strong to go together. Before I learned more about what matches, I'd have to think about this a lot. When I put on a green pair of chinos I'd go to Google images to see outfits of people in green chinos to see what shirt was worn. Or, I'd have to get dressed and then ask others if the outfit looked okay. Looking in the mirror did not give me any sort of intuition about if I had it right.

2.2 The thought process now

Now, once I know what occasion I am dressing for, I can put together an outfit that matches and fits the occasion. Similarly if I see a shirt in my closet that I like and want to wear, it's easy for me to find matching pants/shorts and vice versa. I still do a check in the mirror to see if it looks good but that check is now useful.

2.3 What changed

To get to expertise, I did research and learned basic rules of fashion. *Don't mix stripes with checkers. Certain colors go well with others. The formality of the shirt should match the formality of the pants.* This was declarative learning.

Additionally, from seeing so many outfits that worked well together during my research and from learning what didn't work by trial and error, I got to a place where I can "just tell" what matches. This works even if I can't describe my exact mental model of why these outfits match or don't. This was procedure learning.

2.4 Redesign

A helpful interface would be an app where a user can enter all the clothes in their wardrobe. After this is saved in the app, the user could enter the occasion they are dressing for or its formality. The app would select outfits that would match and make sense for the occasion. Additionally, if the user had a particular piece of clothing in mind, the app could suggest the rest of the outfit to go with it. A final capability of the app would be suggestions on what to buy to compliment what already is in a user's wardrobe, or to fill gaps in occasions that can't be appropriately dressed for.

With this interface, choosing an outfit would be invisible from the start, even from the point of buying the right articles of clothing. The entire focus could then be put on the event/location the user is going to (the overarching task) instead of what they are wearing for it (ultimately a subtask).

3 HUMAN PERCEPTION

Video games will be the chosen task domain.

3.1 Perception and feedback

3.1.1 Visual

Visual feedback is extremely prevalent in video games. Most directly, player input often has an immediate effect on the game world. For example, in many video games, we use the analog stick on the controller to move the character. This movement can be seen on screen in real time. In shooting games when we shoot we will see our enemy fall to the ground or disappear when they die. This quickly shows us that we have achieved our goals.

Visual feedback is also used to indicate danger and when we are close to failure. In some games our characters have health bars. We see the size of these bars go down as we get attacked and up as we heal. In other games, as our health goes down the screen fills with blood from the corners or the screen loses color. As health comes back, these effects disappear.

A more subtle example of feedback is in the game *Mirror's Edge*, a game about navigating through city rooftops via parkour. In this game the correct path through the rooftops is always red while the rest of the city rooftops are other colors. When we see red, we know we are going the right way.

3.1.2 Auditory

Audio in video games is often used for alerting, or where using a visual cue would not make sense in the context of the game world. An example of the latter is in shooting games. Often, there is a tick sound that is played when a shot hits its target. This is helpful when shooting at a far away enemy, where it may not be obvious from looking at the screen that the shot hit. An example of alerting is the loud alarm countdown of grenades near the character that are often hard to see, but pose imminent danger.

3.1.3 Haptic

Modern game controllers have a feature where they can be made to vibrate if certain conditions are met. Vibration is often used in cases where the character is physically or metaphorically shaking, for example when driving a car that crashes, or the character is under duress.

3.2 New design

3.2.1 Visual

We have peripheral vision far beyond the game screen (VR is an exception). This vision is good for detecting changes. A possible way to affect peripheral vision would be to integrate with the lights in a player's house or apartment. A use for this integration would be during horror games. In-home lights that flicker or go out at tense moments would be a good way to increase immersion by going beyond the screen.

3.2.2 Auditory

Players often use one-ear headsets to chat online with friends. In this setup, players hear their friends through their headset and speak through the attached microphone. Game audio may still come through the TV speakers.

Figure 1 — One-ear bluetooth headset.

This could be extended to single player games. A possible interface would be to have alerts play through the headset. This would make sense to do because alerts are urgent and therefore should be close to the player. They also aren't exactly part of the game world. In this setup, sounds from the game world would exclusively come through the TV speakers.

3.2.3 Haptic

Haptic feedback in video games is typically restricted to vibration. One place where the addition of haptic feedback could make sense is the sensation of being pulled in a direction. For example, if playing a racing game and the player steers sharply to the left, in real life, the character would be pulled to the right. Haptic feedback could be used to pull the controller to the right.

3.2 Other human perception

Temperature sensation could be used by video games. In any game where the climate changes or is extreme, the controller could heat up or get colder depending on the current game climate. Heat could additionally be used when characters are under pressure such as in sports video games.

4 REDUCING COGNITIVE LOAD

4.1 Violating interfaces

4.1.1 Work Training - not giving the user control of the pace,

At my job, the training modules don't let me control the pace of the lessons. These lessons are typically videos interspersed with text. I can read the text portions as fast as I want, but the videos, which often do not contain enough visual information to necessitate being videos, don't have transcripts nor do they allow me to adjust video speed. Since I can't control the video speed, I'm less likely to watch the video as watching at regular speed is not productive if I already know the material.

4.1.2 C25k - not emphasizing essential content while minimizing clutter

Couch-2-5k is a popular training regimen for getting into running. It starts by having a novice runner jog in short intervals. These intervals progressively get longer. I have an app made for Couch-2-5k that tracks my progress and tells me when to start and stop the jogging intervals during workouts. When I'm supposed to start running it notifies me (via audio cues) and when I'm supposed to stop running it notifies me. This helps me follow the regimen.

The problem with the app is that there is a lot of audio clutter. Instead of only pinging me to start and stop running, the app gives too much information via audio cues. At the beginning of a 5.5 minute run, the voice from the app says "start running for 5 minutes and 30 seconds. You are on run 3 of 5. Good job!" This is spoken in a slow voice and overpowers any other audio playing through my phone. This is bad when I am listening to music, as I am taken out of "the zone". When listening to a podcast or an audiobook, it means that I missed 10 seconds of content and may have to go back. The information about what run I am on and how long to go are not useful. One reason I use the app in the first place is to remove the cognitive burden of needing to know those things.

4.2 Redesign

4.2.1 Work training

The first improvement I would make would be to put in a video speed option. Perhaps this wasn't inserted due to a belief that watching at faster speeds makes users miss information, but it is worth accepting that humans process information at different speeds, especially given different knowledge bases. Sometimes, too, I want to have the video go slower if I am a novice to a topic and the presenter is speaking fast.

Going beyond the obvious improvement of allowing speed options, the content creators should add transcripts of the videos unless there is a visual demonstration that makes the transcript insufficient. This would fall under using multiple modalities.

Finally, splitting up long videos into shorter more focused ones or marking places in videos where topics are presented would be helpful. This way, viewers can skip around to watch unfamiliar content and then either skip or watch quickly the content that they are familiar with. One can read a textbook like this if they choose to - this type of autonomy should also be allowed in video watching.

4.2.2 C25k

The main fix for my issue of too much audio feedback would be to get rid of the unnecessary speech or to give options to turn these messages off but keep some sort of short ping that the run is starting or stopping. If for some reason this is impossible, the current audio that a runner is listening to could be paused during the messages. This would mean that a listener would not miss anything in a podcast or audiobook.

Beyond this, it is possible that a runner does find the extra information (of what run they are on and how long they have to run) helpful, but they don't want the audio cues. Run information being converted into visual feedback such as a progress bar on a smart watch would be ideal for these runners. This is letting modalities complement each other.