WORLD USABILITY DAY

Theme for Event:
Mobile Communication

November 11, 2010

Kellogg Conference Center
Michigan State University
MR ELLEDGE: The next person I want to introduce to you is Jason Withrow, who is at Washtenaw Community College. Jason chairs the Internet Professional Program at Washtenaw, and he has more than ten years of experience in the web industry as an information architect and as a business systems analyst. He’s also was a classmate of mine at SI and a great guy to work with. He’s also the founder and president of Usable Development, LLC, which focuses on user experience and web design consulting, development, and training. Jason will be talking to us about best practices for mobile application design. Let’s welcome him.

MR. WITHROW: 2:49:40 – 2:51:05 – sound cut out. ---
the less capable devices to the more capable devices, kind of work our way from the, if you will, the older technologies to the newer technologies.

There are many ways to look at this space. I tend to think of it in terms of really two axes. You’ve got capability and you’ve got display size. Really, over time, both of them go up. So, as the years progress we have more capabilities, we have better processors, we have more RAM, we have local storage that maybe wasn’t there before, you know, mobile-focused operating system that’s really good for that, and then, of course, display area also grows. I
guess I would say the one quadrant here that’s just kind of dying off would be the upper right quadrant which would be a not very capable, you know, PDA. Those you kind of see going away because you have just other options. If you want a bigger display area you can get the iPad now, or something else, a lot of things like that coming down the pike. And if you want to have a more portable device you can go with any of the smartphones that are out there, and there are just so many of those.

So some of the constraints. Number one, obviously is the screen size. I want to show you some numbers here so you can get a sense of how much it varies. I don’t even list out here the iPad, so going beyond the iPhone you’ve got the iPad which is even a lot more space. I just want to highlight that this is a huge range and that one of the things that’s always true about user experience is what’s the best answer - it depends. Well, that’s extremely true in mobile. It’s probably more true in mobile than on the desktop. Also, it depends in different ways. Okay, well, what do we want to show? Well, it depends. We’ve got 100 pixels. That’s not much. Or, we’ve got, you know, 480 pixels. Oh, that’s luxurious, we’ve got space, boy, the options. And then you have a tablet that’s even more. I’m going to highlight that even though you have maybe more
space on the iPhone, you end up with a zooming environment, and when you’re zoomed in one of the issues there is you’ve got great focus but you’ve got no context, you have no idea where you’re at in the bigger picture of things. So, there’s always tradeoffs and that’s where the “it depends” happens. I mean, you might not have that zooming on a mobile phone that’s just a mobile phone, but then again, you know...so you do trade it off. More space, if it means zooming, you lose that context.

Software is always a constraint, and really I think we’ve seen this most clearly on the Apple track of denying flash on their devices, and we see that most clearly there, but any plug-in, whether it’s Overlight or Flash or something else you’re going to deal with, at some point, a question of is it even on there, do we even have support for that. So, that’s always a strong consideration. JavaScript is a core part of the web. It’s been for years. With every passing day or month, it’s more a part of what we have. HTML 5 relies strongly on JavaScript to do what it needs to do. So, the future of the web is when more JavaScript is really there. It’s not going away. And some devices lack support for it. They just don’t have it. Those are your lower-end devices, and it’s just going to depend on when you bought it, what generation it was of the
device, and frankly, just what the developer, the person who wrote the – or created the software for the device is going to do.

Local storage is always a concern. You have to ask yourself when you design something are we going to be able to store things at all on the device, period. Can you do that? And, one of the things that we wanted to think about is do you want to target your application broadly or do you want to narrow it down to one device or one sort of line of devices and limit it in that fashion but yet, you know, knowing that they can store things you can then take advantage of that. The broader you make your base of sort of devices, the more tradeoffs you have to make, and really, you end up in kind of a lowest common denominator mode, or you end up doing a significant amount of graceful degradation where some devices just do a lot more with your application than others do. And then users experience it on different devices and think, wow, I thought it saved that stuff on my, you know, Android, and on this one it’s not. Oh, uh-oh. So you have that dilemma as well. And these are not necessarily bad things. They’re just decisions you have to make and those are the tradeoffs that you encounter.
There are a lot of micro browsers out there. It just kind of depends on, I guess, what the device is. Opera has been very successful here with the mini and the mobile, and of course there’s mobile Safari, mobile Firefox is in beta at this point – maybe it’s still in alpha. So, that introduces additional sort of considerations that just happen. You have to be mindful of what’s out there. I guess if you really hate browsers like IE6, you should be happy. It’s not going there. It’s nowhere near it. So, I guess if you are sick of designing things that gracefully degrade for that browser you can be happy. You may not like anything else going on because there’s other tradeoffs and chaos, but at least that decrepit old browser is not coming along for the ride.

There’s always input concerns. Is anyone – if you’ve ever had to write something that’s at all lengthy on a mobile device with a small keyboard or a virtual keyboard, you probably found that it took you a bit longer than maybe you were used to. That’s just one of the considerations. And, of course, this then becomes part of your user interface design process. Okay, do we want to have people typing in things, or do we want to give them buttons that allow them to do that in one press. So, of course, with the great move toward touch screens, you have the option
now of just having, okay, we’re going to have a button for these very common commands, there are buttons for those, and eliminate the typing, because typing is extra keystrokes to even get the keyboard to show up if it’s a virtual one, and then just all the details you have to do with that.

I don’t want to overlook the accessibility implications of this. Obviously, touch does bring in accessibility concerns for users who have difficulty with detecting the amount of pressure they’re applying and how much pressure is needed. If they have difficulties in that area they’re going to maybe not be able to press forcefully enough or press too forcefully, and what might that have as far as implications. And then if the individual has difficulties just getting one finger, they tend to have multiple fingers that touch the screen because of just the impairment that they’re encountering or living with, that’s going to impact touch as well. So, if you’re thinking about the accessibility implications you always have to consider that touch interfaces do have those tradeoffs, those concerns. I do think that touch can, in some ways, be better than your typical keyboard set-up because has anyone ever tried to scroll horizontally, you know, on a typical sort of interface? You have to scroll down and get
the scroll bar and go across. Well, touch, you just move your finger. It’s great. So I’m not going to say that touch is bad or anything like that. It really depends on the situation and what you need to do, and sometimes it will out-perform other ways of providing input. And, of course, the mouse is gone. So if you’re going from a desktop set-up to a mobile set-up, all your on mouse over stuff, you got to re-think that. You’ve got to go away from that. Frankly, the web prior to mobile really becoming big, has been a very mouse-centric environment. You’ll still see a lot of websites that don’t do keyboard support adequately. So we really have to rethink how we move forward from a mouse-driven environment into one that’s going to be touch based.

Finally, bandwidth and connections. I mean, I think this came up earlier when we were talking, there was the talk about essentially how many SMS messages do you want to send in one period of time, you know, you don’t want to kill the server. Well, you know, it’s all about bandwidth. It’s about people wanting to pay for those texts, or just frankly people having not the best connections, period, hitting the usage cap and other considerations. So, I think that with mobile, because it’s a very diverse market, not only the devices but also the providers, it adds
complexity here. It’s not like a typical sort of wired set-up where they have Internet connectivity, and in most parts of the world at least that, you know, around here there’s no limits on that. Now, it’s possible that in other countries they would hit a cap, but not around here. With mobile, they may have a different provider than the person next to them and they have very different cost structures, and frankly, one of them’s on a cellular network and one of them is on WiFi, and they’re getting very different performance from that. So, just things to be aware of.

So, we’ll talk first about the low-end stuff, the things that are really, you know, just older technologies. We’ll talk mostly about the cell phones. The older PDA’s you don’t see as much anymore because they frankly have been kind of supplanted and replaced by smartphones because on an old personal digital system you couldn’t make a phone call. It wasn’t meant for that. And then the iPhone arrives and changes the whole game. So that kind of takes you into a more capable environment.

You’re going to find that with these low-end devices you kind of have, at least these days, your typical web building technologies, HTML, CSS. You might still be developing in wireless mark-up language and if so, that’s
great, it’s just a matter of determining which version to send to the individual. And, honestly, if you’re targeting a lower-end device many times you’re kind of isolating those to a specific address, like a. mobi address, or you know, mobile.something.com, or m.something.com. Or, you’re trying to make just one type that does it all and have a handheld style sheet. So, essentially these are the styles for these devices. The thing to keep in mind is that really for low-end devices you’re mostly just hiding things is what it boils down to. You’re not going to show certain stuff to people who have 100 or 200 pixel-wide screen. That’s kind of your goal was, let’s just eliminate things from view. I wish we could do more. I wish we had a more substantial way of changing what is going on there, but because we’re using cascading style sheets, you pretty much are there to, uh, you know, remove things that otherwise someone on a desktop would be seeing. And, going back to the bandwidth consideration, a big problem is that those people will still download all the stuff that you’re not showing them. So, even though you’re hiding half the stuff that maybe the screen display is showing on a desktop or a laptop, they’re still chomping through their bandwidth and experiencing that lag.
You’ve got to focus on tasks here. I was really happy when Chuck was saying they’ve got an application for this and an application for that because that’s what you do. You’ve got an app that does one thing, or maybe does two things, but it doesn’t do 12 things. It doesn’t attempt to do — especially on these sort of devices, a web-enabled cell phone, doesn’t attempt to do, you know, a dozen various tasks. It’s not going to do any of them incredibly well. It’s definitely a case of diminishing returns. The more you add to it, the less you get for everything you add. It becomes a negative situation. You start to lose after awhile. So those are things to be mindful of.

This is an example of a handheld style sheet operation. This is the rather old website for the program that I teach, and we’re going to do a re-design fairly soon on this, but this is a handheld style sheet. This is Opera Mini showing how it looks for those low-end devices. And, of course, you’ve got to skip to link. This does some thing right. I mean, again, a re-design would be a good thing. So, that’s certainly coming. One of the challenges, though, that you see is that even with this every page looks the same on your device because if you only see so much space, when you go to a new screen it’s really hard to tell that you’re on a new screen, like that
you’ve gone from this page to this page, because, frankly, when you only have so many pixels, the top of the screen, the fold, which you can see without going down, it looks the same. So, it’s kind of a design challenge for all of us. How do we make that different? How do we identify very clearly that you have moved to a different page, versus you’ve reloaded the same page, and really give them a sense of where they are because they don’t have that, it’s not clear. So, this is a very generous screen shot. This is showing a lot more than what someone who is actually on one of the devices would end up seeing.

You definitely want to cut down the clutter. Depending on the display area, you may actually end up reducing a white space to really almost none between paragraphs, where you are very much tightening up things. And it’s because as soon as you see a line of empty space, you might assume that was done, that you basically reached the end of the content, because you have to keep scrolling to see even more lines of text. So, any empty space could suggest that you’ve finished. So, for some users they might not even go on and keep looking. And then I’ve talked about this, and watch out for the graphics being just too much.
The older PDA’s, I’m not going to say much about. This is the only slide I’ve got on it. I mean, to be honest, I don’t view them as a target for me when I’m doing anything in this space anymore. Frankly, if someone has gone to a personal digital system, they’re simply more capable now. They’re going to have some sort of additional communications built in. Probably, they’ll have a phone as part of it, or just something else. And, frankly, they’ve just evolved, they’ve gotten better, more than what they were years ago.

So, let’s talk about the smartphones, and then on from that to the tablets and what you can do there. With these, you really have to choose in the implementation where you want to go with this. You can stick with kind of the web stuff and stay with that stack of technologies, and that’s great. You can certainly do that. Or, you can go with something written in something for Android, and it runs in the Android operating system, or it’s written for Windows, or it’s written for the IOS. It just kind of depends on which way you want to go with this. I would say that if you are a web developer, you do want to be certainly mindful of what’s happening on sort of the traditional website of stuff, which is CSS Media Queries. And the idea here is that you write code that essentially asks the
device what it can do and how it’s set up. So, for example, you can essentially query or ask what the display size is, and if the display size, the region that is visible is greater than this amount, use these styles. If it’s less than this amount, use these styles. And that’s the idea. You can, you know, inquire about – there’s about, I think 12 or 13 different things you can ask about for the device ranging from display area to color depth to just various other aspects, you know, the aspect ratio of the screen, and then make adjustments there. I think the big challenge you have with this first approach, though, is that you’re still, at best, just controlling styles and controlling just how things look, and you can’t really make substantive, huge changes with just CSS alone.

How many consider yourselves in that web development camp? So, a fair number of you, probably at least maybe 25% or 30%. So if you haven’t gone there yet, begin to explore the Media Query side of things because that is something that you can use on your main website. You can have one site or one URL that everyone can go to, and then based on how these queries turn out, people see, at the same address, potentially very different information and things are displayed differently, and tailored to exactly what needs to work for that device. It may not get you all
the way there. You may still have to have some server side logic that checks the device and sees what device is saying it’s accessing the page, and do more changes, but this is a good start, definitely is.

For option number two, really the challenge there is that if you design for one platform you’re really pretty well designing for one platform. What you build here, to port it over is going to take some time and effort to do so. So, those are the challenges that you run into.

You definitely want to streamline the task completion. The situation changes for smartphones compared to – it’s like a web-enabled cell phone, because, let’s face it, on a smartphone you probably have, I don’t know, how many applications? Ten, 20. You’re phone is ringing, buzzing, things are dinging because emails are coming in. When you design something for those devices it’s no longer just a people are doing this one thing and this one thing only. That’s how it was with the old web-enabled cell phones. You were on one task, one path. With the smartphone or a tablet, now you have the email dinging, the phone’s ringing, all sorts of stuff is going on potentially. Other applications are getting feeds and are kind of bouncing or alerting you. I hate to say it, but in a way your application is in a big competition with everything else
there to keep their attention, to keep their focus. It’s a hey, hey, hey, I’m here, you know, keep your focus on this. You may not like thinking of it that way but if you don’t make yourself known, other applications will be used in favor of you. In other words, you’ll kind of shunted off to the side as they look at whatever else is taking their attention. I would say that if you’re familiar with task analysis and also what’s called GOMS, analysis, Goals, Objects, Methods and Selectors, do those things. Effectively with task analysis is doing is having you map out the process with which someone does something, and then studying that, studying all the alternate pathways that they can take, and then optimizing that. And GOMS is even kind of like a lower level of that where you’re looking at, in some case, just performance, what takes less time, what takes fewer movements, what is simply a more optimal way to accomplish something. And then changing the user interface, the interaction design for your application or your website to support those things. So, this is something where I think you really have potentially just a lot of analysis that can be done, and again, keep it focused and make it efficient. So, these are core themes, very, very important stuff.
Now, I think it was mentioned earlier that, you know, pretty much everyone here has some sort of mobile device, and I fully, you know, I guess from my perspective it seems like these devices become extensions of who we are. They really kind of help define us, and frankly, people feel lost without their mobile device. So embrace that. Keep that in mind, that people like their toys. And they like their toys if their toys are basically aware of their preferences. If the toy is --- you know, I say toy --- if the really expensive device in their pocket is set up for what they want, for their last search, for what they search for four days out of the seven days out of the week. Keep that in mind, that people don’t want to approach your application as if they were a stranger every time they go into it. They don’t want to --- I mean, that’s kind of how the web is most of the time. You come in, you know, nothing is really known about you, and then you have to log in to be known. Well, with your device, with your application, because it’s running there on whatever they have for the smartphone or for the tablet, you can be storing information locally because they have those capabilities. And then you can go from there. You know the preferences, and trust me, this is how you’re going to get loyalty to that device is that it works for them. As
soon as they trigger the application it’s got their most frequent actions right there. You can certainly allow them to personalize the choices inside the application, but I would say as a first step, as a first pass at it, do the customization yourself. In other words, have the application automatically remembering what people are doing so that you can basically facilitate those things down the road. So I’ll give some examples, you know, auto loading things and so on.

You want to make sure that whatever you’ve created works, if the person just has literally a single finger or a thumb to work with it. If you’re designing an interaction set-up that is going to involve two hands and a nose, you know, you’ve gone too far, it’s not going to work out for you. With the smartphones in particular people are holding it and also trying to manipulate it sometimes. I mean, it’s just --- who knows what’s going to happen. And this kind of goes into just how modern life has been changed by these. People are multi-tasking and they’re trying to do so many things all at once, and they only really have, you know, again a thumb to work with. So, keep that in mind. Or, maybe otherwise, they have only a finger that they can move and make selections. So, just try to keep that in mind. Limit the need to do a, you
know, keyboard usage. Limit the need for that. Don’t require any sort of multi-touch thing if you can avoid it. It’s just going to be problematic. It just will.

You may have heard of Fitts’ Law. The idea here is that the time it takes someone to acquire a target, to hit something, usually this is in the desktop world, it’s time to mouse over and click on something, it’s really based on how far away it is and how small it is, or how big it is. The further away, it’s going to take them longer. The smaller it is, it’s going to take them longer. You have to slow down and kind of adjust your speed and your timing and just hit it right. Well, on a mobile device, this still applies, but the rules change a little bit. The problem is really more one of --- it’s not so much distance, that’s really not the issue, it’s mostly just size and how big your finger is. I mean, how big is a mouse cursor? A little point, right? A little tip. I got kind of big fingers. I can’t do that. And my nails aren’t anything. So I can’t, you know, I can’t be a mouse or a cursor. I can’t do that. So, how this affects or design process is we have to look at all the things that are around the target and where the person could potentially on accident hit something next to it. I mean, this law still applies with some kind of caveats and modifiers. We’re still
having to adjust performance, and frankly, motor behavior, but now we’re in a smaller space, and frankly, our input device is not as precise as what we used to have. So you have to really get that button just right. It’s not only size, it’s also just if there’s clutter, or is it pretty much a clear zone that you have, and frankly, a lot of them are a big enough region that can be activated. So, if you have a slider bar, make it a big slider bar, make it one that, you know, if you’re not quite perfectly on it you can still get some responsiveness to it. So the clutter is a big part of this problem.

And then we’ve heard this already today about geographic location. This is what can make your application better than the desktop version. How we can get people to embrace it and work with it is you’re giving them location, context, specific updates, and value. So, that can be really, really helpful, very helpful. So, these are things to just think about, keep in mind.

Once you get into the tablet range, at this point this is great. I mean, we’re not yet to the netbook --- well, actually maybe we are, because some netbooks are pretty darn small, but we’re getting closer to that netbook laptop range, and that really does change the game. I mean, the big difference here is that you are able to carry it around
with you much more readily than a laptop would be. And when you compare tablet design to, let’s say, just a smartphone design, the choices you get to make change. With a smartphone, it’s very much about --- let’s say you go from portrait to landscape. What do you do? You typically re-flow the screen. What was once narrow is now wide. That’s kind of your smartphone experience, because you don’t have that much space even with a smartphone. With a tablet, when they move from portrait to landscape, you can change the whole interface. You’ve got room now. Boy, landscape, you know, this is luxurious, you’ve got all this room, this is great. And so the options available to you, once you move into this range, are much greater. You can actually change the interaction style between landscape and portrait, and you weren’t going to do that on a smartphone, you just didn’t have enough room. I know here drag and drop becomes a possibility. You have that option now. Trust me, you don’t want to drag and drop on a smartphone. Oh shoot, I missed it, well, try it again, oh, darn it. You could try it but I wouldn’t recommend it. I mean, anything is possible but it just may not be advisable. With a tablet you’ve got the space, you’ve got the room, so when you’re design processes you can say, well, now we actually have this option available to us, do
we want to pursue it. It’s a touch screen, so that’s going for us. So, now we have the space.

Landscape is really kind of the place where you can get so much value from having a tablet display because that’s where, you know, most of your layouts you’re going to find it being in multiple panels or columns, and you finally have room. And if you really want to maximize your room, you can even have panels that flip over when you touch a corner or something, you can have the panel flip and give even more options. So, you’ve got all of these potential choices that, frankly, on the earlier mobile devices their less, smaller screens, less capable ones you just didn’t have. You’ve got expanding and collapsing panels, and what I would say with this, obviously today a big theme has been research and testing. Test like crazy. See what works. You don’t want to be hiding a critical option and people not know they can flip the panel over to get to it. You don’t want to collapse something and have the person not know they can expand it back up again, that it was actually shrunk down. But landscape gives you a lot of opportunities, and frankly, people expect that your designs will adjust between portrait and landscape because it’s been that way for awhile now, and it’s become pretty well just the expectation that if you’re on a mobile device
of any caliber, that moving the device is going to result in a change in the appearance.

You may also be dealing with cross-panel interaction stuff. So if you have drag and drop, you have a cross-panel situation. So, when your specking out the design, figuring out what to do, testing it with users, you really want to look very carefully at how drag and drop is going to work if you have a multi-column, multi-panel setup; what is not allowed, what is allowed, and how do you indicate that really clearly to people so that they know that this is not a valid action, or how do you even prevent the problem in the first place.

The last thing is portrait. Portrait is tough because, you know, let’s face it, a lot of our applications do better having horizontal space than having vertical space. It just simply does not give you as much of a, I guess you’d say just options. You may end up switching what were previously panels that would flip to being pop-ups, so when you hit something, another sort of little light box comes up with choices in it. Frankly, it feels a lot more restrictive, but people will be operating in portrait sometimes and not always be in landscape. So, you have to consider design for both of these situations. And, frankly, the tablet gives you the room to make these
different experiences, and that’s kind of a different and interesting potentially valuable option to have.

Smartphones, not really. Again, smartphones are mostly just changing the --- you heighten the width of existing elements, and maybe hiding an element, hiding some part of the screen when you move between these. Tablets, you know, you’ve got more space but you still have the portrait orientation limitations. Just things to be aware of there. That’s actually the last slide.

I guess that’s it for my talk today. You’ve got choices, you’ve got lots and lots of options. You can design, if you’d like to, a one size fits all application. If you’re going to do that, I would say go down the HTML CSS Media Queries route because that’s going to allow you to have support on a pretty wide range of devices. If you --- let’s say, on the other hand, want to make a really powerful, optimized app, you may end up creating something in or for IOS or, you know, Android. You can certainly do that. And in that case it might just be sort of a smartphone versus tablet discussion you’re going to have with developers throughout the design process. So identify on this where you fall, where you really want to go, and if you have the analytics, look to see what data you’re getting as far as people coming into the site and what
devices are really in that mix. And from there, you can just move forward.

Okay, are there any questions you have for me? Yeah, in the back.

MEMBER OF THE AUDIENCE: (Not audible enough to transcribe.)

MR. WITHROW: Sure, so the two questions were, has accessibility law, does that encompass mobile devices, and then two, are people developing mobile only. So, I will tell you I’m not a lawyer, I’m not legal expert. What I have seen --- I’ve not yet seen any lawsuits that have been about mobile versions. The lawsuits I’ve seen have been about the regular laptop, desktop experience. But I will tell you that Section 508 of the ADA doesn’t distinguish between --- I mean, it’s just for information technology. In that case, this is included. Mobile is just the same as any other device you’re using, whether it’s a laptop or a desktop computer. So I have not heard of anything or read of anything that would exclude them from that. So I think they’re going to follow the same rules. I would say, though, that one of the challenges you face with that, and one of the complexities, is with a laptop and a desktop situation the input devices are pretty standard. You’ve got a keyboard, you’ve got a mouse. Well, there’s an
interaction on the mobile web between what you as a developer have created and the device the person is using, so who’s actually responsible could be a mixture of the developer and the device manufacturer. So, when something is not accessible it’s kind of a shared responsibility there. So that, I think, would complicate things. We don’t have that level of --- we just don’t have that problem on the desktop usually. It goes straight back to the creator of the website. They’re the ones who get sued. Mobile devices, there’s a hardware aspect to it.

The other question was about are there mobile only, you know, sites. There’s certainly plenty of mobile only applications, but as far as a website being just for mobile, again, people make mobile versions of sites, but they usually will have a desktop version, or they’ll make one version which is for everyone and they’ll use Media Queries to style it differently for different devices. Now, I would say that one of the sort of mantras in web development recently has been why don’t we make mobile be the first thing you design for, so kind of design for mobile from the beginning, and then everything else kind of flows from that. So I’ve been hearing that a lot recently. It’s an interesting idea. I mean, you’ll start with mobile and then let the rest unfold. I guess I would say if it’s
an ideal world, design from mobile and make it a universal
design for everyone, and then let it all unfold.

Other questions?

MEMBER OF THE AUDIENCE: (Not audible enough to
transcribe).

MR. WITHROW: Correct. If they store cookies, if, you
know, they allow that. What you see a lot now, like on
something, for example, on the iPhone you have mobile
Safari. They have cookies as well as a local database
option. There’s actually a rich database you can store
information in. Now, unfortunately what’s been happening
is that we’ve seen that advertisers are using that to
profile people. So there’s always, with any technology,
positive uses and then uses that may not be viewed as so
positive by every person who encounters them. So, yeah.
But cookies would be the start, and then with HTML 5
there’s a local database part to that that mobile Safari
supports.

Other questions? All right, so thank you all for
coming, and I guess it’s probably time for lunch, right?
There we go.