WORLD USABILITY DAY

Theme for Event:
Mobile Communication

November 11, 2010

Kellogg Conference Center
Michigan State University
MR. PIERCE: Well, it’s time for the final speaker of Michigan’s World Usability Day event. It’s time for the final speaker, Nick Kwiatkowski. Nick is a telecomm engineer at MSU’s Telecommunications Systems Department. In addition to his responsibilities on campus, Nick also runs the Michigan Flex (sic.) Users Group, and is an Adobe Certified Professional for Adobe integrated run time for both desktop and mobile devices. A long certification there. Nick will be showing us how to easily create beautiful mobile applications for Android and Apple devices. Please welcome him.

MR. KWIAKTOWSKI: Thanks. So, everybody having a great time here? I’m the last presenter here so I’m surprised to actually see this many people still show up, especially because lunch wasn’t provided. Lunch was good for those people who stayed? Good.

So I’d like to start off my presentation here like asking you guys a couple of questions, and if you guys have your cell phones since everybody has a cell phone here today, I want you to call or SMS these two numbers with responses. So, the first phone number is 618 1032. And I just want to have you guys tell me what type of smartphone or device that you have, number one for non-smartphone device, and number two for an IOS device, three for
Android, four for Blackberry, or five for Windows Mobile or any other symbian device like that. And then the second question which has a phone number of 618 1167, tell me if you’ve ever developed a mobile application before. One for yes, or two for no. So again, you can either SMS the number or you can just call in and hit that number once you’re connected.

So while you guys are actually answering that, a little bit about me. I work for the MSU Telecommunications Department over here on the campus of Michigan State University. Yeah, those results are actually live, so... So, I currently manage the telephone system and the cable TV system here on campus. I also run the Michigan Flex Users Group. The Michigan Flex Users Group is an Adobe certified user group where we talk about things like Flash, Air, developing for the three screens, so that includes desktop, mobile, and for televisions. I myself am also an Adobe community professional. A community professional is somebody who has been sought out by the Adobe community to actually represent some of these products.

Looks like we still have some coming in here.

AUDIENCE MEMBER: Can you give us the two questions again.
MR. KWIATKOWSKI: Sure, the first question is what is – if you currently have a phone with you, what type of phone is it. Number one is for non-smartphone device. Number two is an IOS device, so that would be the iPhone, the iPad, iPod. Three if you’re using an Android device. Number four for Blackberry. Number five for Windows Mobile or something else. The phone number for that is 618 1032. And then the second question we asked is if you’ve ever developed a mobile application before. The phone number for that is 618 1167, and the one for yes, two for no. We’ll let that go in there for a second longer as well. By the way, this over here is actually written in Adobe Flex, using Adobe Cold Fusion in the background. This is actually a server sitting in my office right now that you’re actually calling into.

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

MR. KWIATKOWSKI: Yes. So for all of you who actually don’t use a smartphone, it looks like 21 of you or so, you guys didn’t raise your hands earlier. Before, the last presenters actually thought that everybody used a smartphone.

MEMBER OF THE AUDIENCE: (Not audible enough to transcribe).
MR. KWIATKOWSKI: True, okay. So, it looks like most of the people here are actually using an iPhone. That doesn’t surprise me in a tech environment. That is actually very typical that most people actually use an iPhone. Second, obviously is going to be Android. I’m actually surprised to see that Blackberry isn’t higher than it really is on that list. Typically, we see about 40% of the people using Blackberrys, with the remainder split usually between iPhone and Android devices for most tech conferences or most things like that. So, this is a little bit skewed. Also interesting to see that we have about 25 people here that said they actually have developed a mobile application before. That’s also usually very unusual. A lot of times people, especially in the usability world, at least when I checked last time it was maybe like 5% or 10% of the people who actually developed a mobile application, where almost everybody else will say, mobile what?

So going right along. Why should you really develop mobile applications? Some mobile and portable devices are kind of the wave of the future. As we heard in some of the prior presentations, even the state of Michigan has realized that people are abandoning actually using their desktop PCs or they’re using their desktop PCs even less, and they’re actually using these mobile devices that are
with them all the time. Pretty much from the stats I’ve seen, 4.6 billion cellular subscriptions worldwide. Pretty much every continent has at least one cell phone tower on it including the big Antarctica and places like that. Of those, 3.5 billion, that’s with a “b”, are capable of text messaging. So that includes pretty much every phone out there. I can’t find a phone that’s been manufactured in the last 15 years that is not capable of text messaging. Not everything is capable of actually doing full-blown applications, or even browsing the web, but almost everything is capable of using text messaging. Within the United States, about 257 million data plan holders is what they’re kind of estimating right now. That’s kind of a fuzzy number because a lot of the manufacturers and cellular companies are not really releasing how many data plan holders they really have. Over the entire world, we’re expecting about one billion people have a data plan of some kind which allows them to get onto the web. Regardless if they have a device that’s capable of the Web or not, they actually have or are subscribing to that service. One thing to note with mobile devices as well, you know, I’m kind of selling this to people who already realize that they should be making mobile applications and mobile websites, is the significant face time with mobile
applications, both with dollars and interest associated with them. You may have an application that’s sitting on your computer at home. You may use it every so often. With a mobile device, I always have this device with me everywhere I go. I go to the Twitter website on my desktop PC, yeah, a few times a day or something like that, but I’m checking Twitter on here like seven times a day, eight times a day. It’s always running. When somebody sends me a message, you know, it actually pops up on here. This is something I have with me all the time, even though I’m not in front of a computer.

Unfortunately, up till now, the mobile market was very, very costly to deal with and also very difficult to deal with. As you know, we heard in the last presentation, each browser on mobile devices is different. A lot of them don’t even support JavaScript. JavaScript is kind of like the driving force of HTML 5 and all that. Less than 30% of smartphones today have actually --- can support any aspect of HTML 5. So, despite what the famous turtleneck person said where, you know, everything is HTML 5, that’s not really the reality yet. In fact, 30% is a very, very low penetration rate, so something you can’t even rely on yet. Many phones don’t even support HTML. A lot of the web-capable phones, for the people that raised their hands that
said they didn’t have a smartphone but they actually support web, use a protocol known as WAP. WAP is an HTML-like protocol, however, it uses a series of what they call pages and cards where you can actually switch between cards and relies on a service provider that actually downloads ahead of time, and it doesn’t make it so it’s an interactive application. Many of the applications themselves are specific to each platform. So, I’ll get into that in just a moment here. Additionally---

(Sound cuts out – 3:36:45 – 3:37:41)

---actually checking for websites. Let’s say the state of Michigan. If they really wanted a whole lot of penetration, people are actually searching for state of Michigan applications using the iPhone App Store, using the Android Market, or things like that, before they’re actually checking to see if there’s a website. That’s completely backwards to what people are used to as far as the developers in the world. So, of course, there’s a lot of fragmentation within mobile devices as well. If you’re trying to develop for native applications, things like the Android, you have the native Android code. You have Flash and Flashlight. You have JavaScript that you can actually develop applications for. For Apple, you’ve got the native Apple code, also known as X code. JavaScript or HTML 5 for
RIM otherwise known as Blackberry, you’ve got Java applications or the native RIM applications. Windows Mobile, you can either use the Windows Mobile CE compiled native applications, or you can use Silverlight. You’ll notice that there’s very few things that are actually touching all those different types of devices. However, I’m going to be talking about a framework, and I’m going to actually show how to develop an application using a framework called Air, which is, as you can see on here, is compatible for a lot of devices. It’s available today for Android. It’s available today for Apple, so that includes all the Apple devices. For Blackberry and for RIM it’s coming out extremely soon, so they already have betas out there today that you can download. They’re going to have more applications --- or more support for that very shortly. And Windows Mobile, particularly the new Windows Mobile phones that were just released yesterday, there’s going to be compatibility, in fact there’s some work on that today.

I’m going to talk a little bit about the Air run time, or the Adobe integrated run time. Adobe has published an application framework that allows developers to create applications that run across all what they call three screens. So, that includes desktops. So you can actually
write one application that will work on both Windows, the 
Mac OS, and Lennox. And mind you, this is one 
application, the exact same code. You actually deploy a 
.air file and it will run on all of these different 
operating systems. Similarly, you can also take that same 
code base and make an application that works on mobile, 
including the Android OS, IOS, Rim, and coming soon Windows 
Mobile. Additionally, you can also take that same code 
base and develop it to televisions. So, one of the things 
that they just announced even two weeks ago at the Adobe 
conference was that Broadcom, Tivo, and Motorola are all 
supporting this run time in their future versions. In 
addition, one thing I’ve actually been able to play with at 
home is I have a newer Samsung hi-def TV. I’m actually 
able to write applications that run on that TV. For 
example, I just wrote a really quick application for the 
Alumni Association here that allows me to browse all their 
YouTube channels and just using the remote be able to 
switch between those without actually going to the YouTube 
page.

Air allows you to reuse your code to create 
applications over multiple platforms. I do put the double 
asterisks there because it’s not always --- doesn’t make 
sense. It doesn’t make sense to have the same user
interface for a phone as it does on a desktop as it does on a TV. I mean, for one thing, the user interface for those is completely different, the input mechanisms, I should say. For phones, you expect the thing to use things like gestures. For television, all you have is a remote. There is no touching the TV, or at least they better not. But it still allows you to use a common code base. So, if you have a whole bunch of complex business logic or code that allows you to interact with the web service, you can still keep that, you just have to make a new user interface.

Tools are available today. So, a lot of these tools I’m going to show you were actually released in April of this year were in beta over a year before that. In fact, a lot of them are included in Adobe Creative Suites CS 5. So, that’s a very popular application suite. I’m sure a lot of you have probably used it, especially if you’ve already bought Photo Shop. The air toolsets themselves, just the caveat of all this, it’s never going to be as powerful as a native application. It will always be more powerful than HTML, HTML 5. However, 90% of the applications out there, you don’t need to be doing like 3D processing, you don’t need to be doing really heavy work within the mobile device or within the television. But it can do just about all of that. One thing I do want to
mention, the PodCast games, they actually did a meeting for me a little while ago. They’re the makers of Plants versus Zombies, and Bejeweled. How many people have never heard of Plants versus Zombies before? If you have an iPhone or you have any type of smartphone, go to their respective market or store and search for it. It is a highly, highly addictive game. Bejeweled is actually what made them their big money. Plants versus Zombies has only been out for a few months. But they said for them to convert a native application from the desktop to the Apple system and then also to Android, it took them one person one full year. So (inaudible) they actually had three or four people working a quarter of a year. That is all that they did was just porting that application over to the other device. Air actually helps that out a lot by allowing you to use the same code base, but not necessarily having to write everything from scratch.

So, I’m actually going to show you how to write a very simple application. If you go to the Android Market right now, if you have an Android phone, you can actually search for Application by MSU Directory. It’s published by myself. It’s not published by Michigan State University, so hopefully people from University Relations aren’t here
and fine me with all sorts of branding rights and all that stuff.

MEMBER OF THE AUDIENCE: Can we tweet about it?

MR. KWIAKTOWSKI: You can tweet about it all you want. So, this particular application, the one that’s actually on the market is a little more polished than what I’m going to be showing here. We’re not going to have time to actually finish the full application here. But we’re going to show like 90% of the functionality, how to actually build that right from here. This particular application, it ties into the MSU Campus Directory and allows you to do live searches of people and actually call them, find out where the location is, all that type of stuff. The neat thing about this entire application is we’re going to be, you know, using a pretty neat workflow on this. We’re actually going to start this application by starting it in Photo Shop and build a fully functional application right from there.

So, a couple of caveats. I’m not going to be dealing specifically with this presentation, I’m not going to be dealing with best practices. We’re going to cover some tips and tricks in creating the mobile applications at the end of the presentation. A lot of them were already covered in some previous presentations today. The final application will not be screen reader compatible. There
are live techniques, and I’ll point out some of them as we do our presentation, but this is more just to do --- I’m going to be building an application in 30 minutes. I’m not going to be able to, unfortunately, to show every possible programming technique that would be required for that. We will also be targeting one screen size and one orientation. I’ll show you have to incorporate those other orientations in there as well. But, yeah, so let’s get started.

So, like I said, we’re going to be using a workflow here. We’re going to start off in a program called Adobe Device Central. Again, this is something that’s included with every copy of Flash CS5 or Photo Shop CS5. Device Central is actually an application that will allow us to inspect certain devices out, actually having them in front of you so you can see what the screen resolutions are, you can see what their support is for video and all this other stuff. We’re going to jump into Photo Shop for a little bit and actually prototype out the application. Then we’re going to take that prototyped application and go into Flash Catalyst, which is Adobe’s full-fledged prototyping application. Flash Catalyst is an application that allows you to take a Photo Shop file, say this is actually a button, and make that live. It will allow you to take something like a box and say, this is actually text box
that people can type into and make that live so you can actually do these things. We’re then going to put it into Flash Builder, which is an application that primarily coders use. I’m going to show very quickly how to hook it up to the web, and I’m actually going to publish that to -- I’ve got three devices up here right now. I’ve got a Motorola, let’s see, I’ve a Droid 2, I’ve got a Droid X, and then I’ve also got a Samsung Vibrant that we can actually deploy that same application to all of them at the same time. Unfortunately, I did update this particular machine to Windows 7 and the Apple Took Kit isn’t readily available for it yet, so I won’t be able to show it to actually pushing out to an iPod, but the same workflow would be the same for that. So, we’ve already went through this right here.

So, we’re going to start off with Device Central. Now, this is a live demo where I’m actually going to be typing and all that stuff, so I may come up with errors and all that type of stuff. Please throw things at me if you see something that doesn’t look right. Ask questions, all that type of stuff. But the first thing we’re going to do is load up Device Central. So Device Central is going to load up over here and the first thing it’s going to ask us to do is, you know, what do we want to do, if we want to
create a new mobile application or things like that. I’m just going to browse the devices right now. So, this particular application right here has profiles for pretty much every mobile device that is pretty much known. They’ve got some dating back about four or five years. So, for example, if I want to look up the Droid 2, I’m going to type in Droid 2 on here, so the Motorola Droid 2. It says here it’s actually be Motorola, and there’s all sorts of other ones in here as well. I can double-click on that and it will show me all the different specifications about that particular device. The thing I’m most interested about is the color depth. It actually supports 24-bit color. The display size, you know, the dimensions. It shows me that it’s on Verizon, and it actually would show which countries it’s capable in as well. What networks it’s on, you know, all sorts of things. I can go to the web and it actually tells me what the user agent is. The user agent is very useful if you ever build a web page and you want to find out if a Droid 2 user actually views your web page. So that user agent over there will uniquely define that particular device on your web page when you’re looking at the server logs. Other things that are important on here as well is it says what supported formats on here. So this particular phone, the built-in browser supports CSS Level
1, 2, and 3. It supports HTML4 and 401, WML, all this other fun stuff, and it also supports JavaScript. If I go to the Flash tab, it tells me all sorts of other things as well. In particular, what’s kind of interesting, it tells me what the input is, you know, if it uses Stylist or if it uses --- for some odd reason, it says this one uses Stylist, which isn’t necessarily true. But, it says it has Excel or ROM support), if it has GPS support, right from the web page and things like that, which again is very useful if you’re ever trying to create an experience for a particular set of users.

What I’m going to do is I’m actually going to create an application for the Droid 2 because that’s the one that’s currently plugged in. I’m going to say here I want to create a Photo Shop file. By the way, you can also create things for After Effects, Captivate, Fireworks, Flash, Illustrator. I’m going to use Photo Shop because that’s what I’m most familiar with, at least for this particular demo right here. Say I want to create this, and it will create. And then it’s going to go load up Photo Shop. This will take just a moment here. Unfortunately, I’d like to say that Photo Shop is the largest application on this machine, but it’s not, as far as the slowest. So here is a Photo Shop file that’s already set up for the
proper dimensions, for the proper color depth, for
everything else just to start creating an application. I
can start creating circles, I can start putting text on the
screen. I can do whatever I want with this. This will be
ready to deploy right to a phone. So for those of you who
actually do prototyping and do device prototyping, instead
of looking up what every single device’s, you know, what
the dimensions are and all that stuff, you can actually use
as work flow, hand it over to your designers and say, here,
make an application for me. Conversely, if you want to,
you know, if you have a banner ad or something like that,
you can use this same workflow as well.

So, I’m going to go --- let’s see here. I’m going to
set this up for application. I’m going to rotate this
right here so it’s actually in portrait mode, and I’m going
to go paint the background and do all that fun stuff. So,
let’s go and make this a nice gray color. Like I said,
you’re going to see me clicking and doing all this stuff,
but, let’s see here. Make it a nice gray color in the
background. We’re going to go and put in a couple of image
assets. So, I’m going to go and put in the MSU top bar
over here. I’ll put this up top, so we’ve got a nice
little header on here. We’ll put the Spartan logo on
there, which I have somewhere here as well. We’ll put this
up here. And you see that, I mean, this is starting to feel like an application already that we’d be used to. This should be no different than any type of designer or prototyper that you would have. What we’re also going to do here as well, we’re going to put in a --- let’s see here --- put in some new text. I’m going to make this white, if it will let me. It’s just a little slow today. We’re going to say Employee Directory, and we’re also going to put in a white area as well. So I need to open up my layers tab here. We’re going to do a new layer, and I’m going to put in an input box area, right over here. So I’ll put that over here. We’ll put a little shade on it just so it looks nice. I’m not diggin’ that my mouse here is on a slant and keeps on moving on its own. Okay, we’ll do an inner shadow on here so it looks like it’s a little input box.

So, this is what we have. We’ve got an introduction to a basic application. Now, in order for a prototyped Photo Shop file to actually work as a real application, one of the key things is we actually have to put each of these elements on their own layers, which we’ve already done. You’ll notice I didn’t put the input box actually in the background and things like that. If we do that, we actually lose the taxonomy of the prototype itself. But
another thing that’s kind of important is we actually need to organize things into folders. So I’m going to go here and we’re going to go and create a couple new folders. I’m going to create a folder for the top bar, and we’ll go and stick all this stuff in here. So that all goes into my top bar. Another thing I’m going to do, too, is I’m also going to create a --- oh, one thing I forgot to do is create an area for the text. So, a lot of text boxes when you build them, have typically like some instructions pre-filled in for them, and then once you click on it that text goes away and you can actually type something. So, for example, I’m going to click in here and we’ll say, last name comma first name, which of course I forgot to change my default color from white to black, so we’ll do that here. So this is what you would typically see. When you first open the application it will prompt you to say, hey, you know, put it last name comma first name. So, I’m going to create a new folder within the top bar and we’re going to say --- this is going to be the input box. I’m going to put that shape and the text in there as well. Now instead of having you guys watch me suffer going through Photo Shop for the next half hour, something like that, to put together the rest of my prototype, I’m going to jump ahead and actually load up one that I’ve already done, for the rest of the
prototype. Because, you know, Julia Childs, you know, look what I’m pulling out of the oven here that’s already done.

So this is what I’ve already come up with here. It should look almost exactly the same. Of course, you know, I set the text a little bit, and all that stuff. I also have a repeating item on here as well. So, all the search results that come back are going to have the Spartan head, the name, and what department it’s part of. What I would really like to have is have everybody’s photo on there, but unfortunately MSU doesn’t actually expose all of our I.D. photos like I thought that they originally did. So everybody is a Spartan head. Another thing is, once we click on one of these items, I also want to show a details page. So, I’m going to load that up here as well. Again, this is all using basic Photo Shop stuff. This is nothing that crazy. So I’m going to show the name, department, email address, the location, and phone number. By the way, all of that information is real. If you ever want to contact me, please do. The things to note with this as well, again I’ve got a details area and I’ve got all the different text areas in here. I’ve got an action bar which is this bar down here at the bottom which has a “call me”, a map, and then back to the last screen. Again, these are just placed items on here. Some of them are smart objects,
some of them are just images that I’ve put in here. And then the top bar is exactly the same as last time. So this shouldn’t be too much of a surprise if you’ve used Photo Shop a heck of a lot. In fact, my Photo Shop skills are not very good at all, and somehow I managed to come up with this. So, I’m sure you guys can come up with a lot better.

So, now we’ve got these two designs. They’re saved as standard PSD files, I’m not exporting or anything like that, I’m just going to be making sure that these are saved. And we’re going to go close out of Photo Shop. But before we do, I’m going to jump back to our Power Point really quickly here and show some of the key points that I want to cover. So, again, key points with Photo Shop, separate each of your functional components into their own layers, and then, you know, into folders. That’s extremely important and that should be important for anyone who deals with Photo Shop. Don’t ever send a developer or anybody else a rasterized version of your Photo Shop, everything on one layer. They will scream at you, they will yell at you, and then they will probably contact somebody else who can actually do the work. One thing to be careful with labels and fonts. Custom fonts look really, really nice. Unfortunately, they are very big and they use a lot of memory when you actually deploy to your device or deploy it
to anywhere else. Another thing, too, is when you actually --- another thing is don’t rasterize your labels and your fonts. That will break all accessibility. Make sure you keep your fonts. Make sure you keep your labels and all that stuff as actual labels and actual fonts. Without that, what will end up happening is screener will say, oh, that’s just an image and it won’t actually display to you what’s actually happening.

Next thing we’re going to do is go to Adobe Catalyst. Adobe Catalyst --- how many people have heard of Adobe Catalyst before? Wow, that’s a lot more than I expected. Usually I go into a room like this and it’s like three, or that 1% or whatever that was, yeah. So we’re going to go into Flash Catalyst. Flash Catalyst is an application, like I said, that will convert a Photo Shop Illustrator file, or, you know, the next version is going to include a lot more different types of file formats, and make that design into a functional prototype. So once it loads up here…. Again, none of these --- my machine is not that quick, but it gives me extra time to ask how everybody’s doing so far. Good? Okay. So, we’re going to create a new project from a design file. So we’re going to create a new project from a Photo Shop file, a PSD, which is what we saved that in. We’re going to go find my file. I luckily
saved that over here. And it’s going to import that. It’s going to come up with a quick dialog box over here that’s going to ask me a couple of different options. Notice that the width and the height are automatically populated in. I do want to set my background to something other than white because when people see white on their screen they think something is wrong. So I’m going to change it to that light gray color, let them know that this is actually what the real background is. I’m going to hit “okay.” It’s going to analyze this file and go create a functional project for us…eventually. Really eventually. All right, here we go. So again, here’s our same Photo Shop file. I’m going to zoom out a little bit. For most of you guys — okay, you should be able to see most of it there. Come on. There we go. Okay, so again we’ve got this exact same file that we were just working with. And just to show you what it looks like when we actually convert this to Flash, I’m going to run this project by hitting the control enter key, and it’s going to compile this and launch it into Internet Explorer. Yes, it will. There we go. So here we go. This is this Flash file. You notice that nothing really works. I can’t click on anything. All I can do is scroll back and forth to actually show the application itself. So let’s make this thing functional. First think
I’m going to tackle is this text box here in the middle, or in the top bar area. You’ll notice earlier when we had that oval area and we had the text inside of it, we’re going to select both of those. I’m just clicking on them and hitting the shift key to select both. And we’re going to convert this into a text box. It’s going to say, here, I’ve converted this into a text box, or a text input, and now it’s actually a text input. That’s all that was required. Now, let me go run this back in Internet Explorer again. There we go. You’ll see here now this is actually functional. I can go here and I can type in stuff. This is coming right from the Photo Shop file, which is amazing. This is technology that really hasn’t been around too long. Let’s continue along here. The next thing I’m going to do is convert the scroll bar for our data grid, so when we have more than one page worth of information, we like to show a little scroll bar on the side to let people know how much information there actually is on the screen. So, I’m going to highlight the scroll bar and what they call the thumb. I’m going to convert that into a horizontal --- no, vertical scroll bar. We’re going to edit the parts. So here I have to actually identify what is the track and what is the thumb. So I’m going to click on what is the track, which is the larger
area, and I’m going to say that is going to be a track. The thing in the middle, which is the thumb, so think of that as like the highlighted area in an actual scroll bar, like this right here. We’re going to say that is the thumb. That’s all that’s required to say that’s a scroll bar. So, now Catalyst knows that as a scroll bar. Let’s go and make the data region over here, like the results that are going to come back as something that’s functional as well. So, we’re going to highlight all these. Come on. It’s not liking something. There we go. Okay, so we’re going to say all this stuff here is all part of that data list. We’re going to convert that to a data list. Now, we have to edit the parts again, and we’re going to have to say what is actually repeating. So what is something that is displayed for every single result? Again, I want the Spartan head, the name, the department name, and then that separator. I’m going to say that is my repeated region. Okay, we’re okay with that. There we go. So, now it knows that that --- all those items actually repeat. I’m making sure I’m not forgetting anything, so I’ve got a list of stuff I’m supposed to show.

I did notice that these items here are a little too close to each other to be comfortably hit with a thumb, so what I’m going to do is actually change the layout and I’m
going to change the spacing so those are actually a little further apart. Go back to my file and designs. That’s kind of what it’s going to look like. Again, I can hit control enter to preview this application, and once Internet Explorer shows up over here, again I’ve got the area over here, and now it also shows the highlight as each of these items over here that would come back from the result. Is that cool so far? Cool.

So the next thing I’m going to do is I’m going to import that second state that we had. So remember we had the search and also the results, and then we had a details view of that same application. That details view shows more of the detail, like their actual phone number, the email address, things like that. It allows you to actually call this person. In this case, you’re calling me seven, eight times, whatever it is. We’re going to go and rename our first state and we’re going to call that search. And then we’re going to rename --- we’re going to create a new second state --- actually I messed that up there. We actually need to duplicate the existing state. So we’re going to go duplicate. We’re going to call that the details. We’re going to remove data grid, our search list, and I’m going to import that second Photo Shop file. So I’m going to go to file, import from Photo Shop, I’m going
to take that view, and go over to the advanced, and have it say I only really need the action bar and the details. Everything else is already in here. I don’t really care about those. So it’s going to go import those. I just need to position them on the screen properly. So this here should be near the bottom, make sure it lines up okay, and there we go. So now I have the search view and I have a details view. For a quick demo, it’s not too bad yet.

From there --- so now I’m going to go create a couple of interactions. So on the data list here, I’m going to open this back up. So when somebody clicks on one of these items --- wait, clicked on the wrong one here. When somebody clicks on something on the data list, we’re going to add interaction, so when they select something we’re going to go play transition to a state, and we’re going to say we want them to go to the details view. So when they click on something or they hit it with their thumb, they’re going to go to that details view. Conversely, in the details view, we’re going to hit the back button over here and we’re going to convert that to a button, so that’s something they can hit, do an interaction and say when they hit that button we want them to go to the search view. Hit okay. And that’s all they need. So, now when we run this application, we’re actually going to make this thing do
something. Do something is kind of a relative term. So we can still hover over these things. Again, hovering isn’t something that really happens in a mobile application. Mobile phones today don’t sense when your thumb is over something. They sense it when your thumb is actually on something. So when you actually go click on something or actually hit something, it’s going to switch to the details view. When I want to go back, I hit the back button, and it goes back to the list. So right here, as far as a prototyping tool, it’s extremely useful. So if you’re, let’s say, a typical designer, you’ve come up with a Photo Shop file, and then you try to explain to the actual programmer and say, when this button is hit, I want to do this, when I click on this I want it to do this. You can actually have it so the application, when you hand it over to the developer that actually does those things, so they actually know what you mean.

Just for a couple little frills here, I’m actually going to show the timeline within Adobe Catalyst, and there’s actually a way for us to smooth the transition between one state and the other. So I’m going to quickly do this, smooth transitions, going from the search to details, and then also in the reverse. What that pretty much does is it figures out what components are changing
and it makes them fade and do all that stuff. So again, really quickly, running this. You’ll see when I click on one of these items, it’s going to fade from one screen to the next. Now just as easily I could have had it, you know, so it slides from one screen to the next. That’s very typical on the Apple and the Android-type operating systems, where if you go from one page to the next it actually does kind of like a slide. This one here was just one button away, so I wouldn’t click that, but you can also do those slides, you can do it so it does like a spin poof thing, or whatever you might have, you know, whatever prototypers and designers want you can pretty much do with this as well.

So I’m going to save this file. It’s mighty fine, if you ask me. And we’re going to save this, and save it as final design, on my desktop, and then we’re going to close Adobe Catalyst, and go into the actual programming I.D. But to have some quick points on here, you know, within Catalyst you convert each item that you want to be active, so think of it as something that the users actually interact with. By clicking on it and going to that HUD, that little box that kind of floats around there. When you’re working with data lists and things like that, convert your scroll bars first and work forward from there.
And then anytime you want to run or preview your application, you can hit control enter, or there’s also a menu item under the file name. Conversely, you can actually have it so these applications can actually be deployed as .swf files. So if you want to show people, let’s say, on the outside, your clients or somebody else you’re working with, you know, what you’re actually working on, you can deploy it as a .swf file, which then they can run pretty much in any browser that’s on the computer.

So now we’re going into Adobe Flash Builder. I’m going to run through this really quickly because it’s just about the amount of time that I have. Adobe Flash Builder is an IDE that’s used by certain developers. In the Adobe world there’s actually two ways --- actually, there’s multiple ways to make flash applications, but there’s two very popular ways. One is using Flash Professional, which is an application that’s been out for about 15 years now, or 12, something like that. That’s where you find a lot of people, especially designers, use to make flash applications. There is a newer application that came out about three or four years, I think it’s closer to four years ago, called Flash Professional. It used to be called Flex Builder. It allows you to create flash applications, so swift applications, by actually writing code. That’s
actually what I’m most comfortable with. When I load up Flash Professional, I start getting really confused, dazed. I go into a corner and start crying, because it is one of those designer applications that I just have no idea how to use. I tell myself I’m going to learn it one of these days but, you know, it still hasn’t happened. But, you know, I’m a coder at heart, so this is what I’m most comfortable with. So, you can actually do the exact same kind of workflow by working in Flash Professional rather than using the actual IDE, but typically you’ll find that a lot of developers, particularly those that are used to the Visual Studio, or any of those types of IDE’s, are most comfortable with this type of set-up.

So, I’m going to do two things. The first thing I’m going to do, I’m going to create a new mobile application that we can deploy to a phone. So I’m going to right click on the Project Explorer, say “new Flex Mobile Project.” I’m going to have to give this a name, so I’m going to call this Demo Directory. I’m going to tell it that I want to use the current Flex SDK and all the stuff. These are all the things that design --- or, I mean the developer would know. I would list my target platforms and then I would say what type of template I want to use. Mobile application kind of stubs out a basic application. If you
start in the work flow where your developer is prototyping stuff for you and then a designer works later on, you would actually start off with a mobile application rather than blank, because I’m going to be importing what I have already, and I’m just going to do a blank application.

Now, one of the things I said earlier that I’m not going to worry about the orientation, so I’m actually going to turn that off here in the application setting. I don’t want it to automatically reorient my screen. Now, if I were to build a real application that I would be publishing out to the world, I would actually make multiple Photo Shop files where I would actually have this different screen orientation and things like that. I didn’t do that here just because of time. I’m going to say if I’m working with a particular server. I don’t really care about that part right here. Then I have to give it an application I.D. The application I.D. is something that is unique to you within whatever the store you’re working in. So, for example, if I were to actually publish this out, I would use an application I.D. of edu.msu.telecom. --- and then an application name. Let’s say directory demo. If I were to publish this, this is what the application I.D. would be to that particular store. It’s very typical that you would use your domain name backwards because, you know, there’s
really only one set of domain names out there. So, for people who are working for MSU, you’d use msu.edu. – probably your department name, dot, your application name, or however you want to segment that out, as long as it’s guaranteed to be unique. I can guarantee you that there’s at least one application out there in the iOS store – yeah, the Apple Store, and then the Android Market that is called Hello World. You would not be able to publish your application with the name Hello World.

So, I’m going to hit finish and it’s going to create a stub application for me. Moving right along here. Okay. Come on. Okay, so this is a stub application. I can go to design view, I could go and actually again, this is kind of set up for, if I’m starting from a fresh application, because I’m going to import something. I’m going to do that right now. So, I’m going to right click here. In the package explorer, go to import, and I’m going to import an existing project, I say I’m going to import that final design I was working on, hit finish. Now, there is actually a bug in this particular version of this IDE that I’m working with right now that prevents me from importing it into the existing project. That will be fixed in future versions. So I’m going to just copy and paste things over. So right click on my source directory here, paste it in
over here, and then I can go close that project because that was used to import stuff. And there is – oh, and I brought it into the wrong place. So, let’s try this again. Paste it. It’s saying that it found some other things. Okay, I messed up. Let’s try it again. So I’m going to copy this, paste it in over here. There we go. Now I can close this because we don’t need that anymore. I do have to fix one little bug that, again, the dirty little secret is I’m actually using a beta version of software over here because there’s one feature I want to kind of show you guys. There are still some bugs in here which is why I’m fixing this right now. So, I’m going to fix this so when it imports it, it doesn’t do everything 100%. But these bugs are actually very well documented, so if you were to use this today, you’d actually know about it as well. Let me go and remove this, and remove this right here. And I’ll hit save. And I forgot one last thing right here. There we go. That should hopefully build everything. We should be all set. So now I’m actually going to go and put this into a debugger, so I’m going to click over here to actually run this project. I’m going to say it’s going to be a mobile application. Okay, let’s try this again. Debug as --- let’s try one more thing over here. Really, this did work last night. So, I go and run the debugger
and then I say I want to either deploy this on the desktop as an emulator, or I can actually deploy it onto a device. So I’m going to say on the desktop, and I’m going to target a Droid 2, and hit debug. And you’ll see a screen pop up here in just a moment that has all the same dimensions as a mobile application. Again, the hover thing and things like that don’t work because they don’t actually exist on a mobile application. When I click on this, it actually switches over, and if I had larger screen, you could see the back button as well. But, just as easily I can go and run this on a mobile application. So I’m going to go run configuration here. And I’m going to say I want to deploy this on an actual device. So, now this is where there’s going to be some trickery over here. I’m going to see if I can load up a web cam so you guys can actually see what’s on the phone because I’m guessing that you guys can’t actually see what’s on this little screen. So right here we’ve got a regular mobile --- this is the Droid 2 again --- and it’s launching right now. And you’ll see on the screen the application is loading, maybe. There we go. So I actually have was a Photo Shop application sitting here on my phone. And when I click on one of these you should see it go right to my screen. So, that right there is a mobile application. Cool, yes?
So, just to make things --- let’s see here. I’m going to make this do actually just a little more since we have -- how much time do we have? Another 15 minutes. Wonderful. I’m going to take this application as far as I can and then once we have about five minutes left I’m going to run through the last of my slides. Just so you guys know, I will be publishing all this stuff on my blog. I will give you my blog address at the end here so you can download this and actually play with it. But right now we’re going to make this work a little nicer. Let’s see here. I’m going to give that text box that’s on the top, I’m actually going to give it a name. I’m going to call it search field. And then we’re also going to add some code. Again, this is all stuff that a developer would actually deal with. I’m actually programming in what’s known as Action Script, which is a language that’s fairly similar to JavaScript. Cool. So then --- I built all these scripts over here so you guys don’t have to watch the pain of sitting there watching me type and miss-spell because I am the worst speller in the world. All right, what happened there? Oh, wrong one. So, I’m going to say when we click on that we’re actually going to go to search field. What is that called? Search field clear. These are just custom functions that I created just earlier. So we’re going to
deploy this onto the desktop again just to see how it works. Save the file here. It’s going to go run it. So when I click on this or when somebody hits the button --- actually clicks on it with their thumb, it will clear all that out so they can actually start typing instead of having to wipe out what was there before. So that’s the first step in making it a little nicer application. Let’s see here. We’re going to do some other stuff in here. We’re actually going to connect this up to the network, to the Internet, so we can actually get live search results. So I’m going to do my macro of demo. There we go. That added a whole bunch of code that pretty much hooks it up to the network. Again, this is all stuff that a developer would do for you. Application complete. Saving it. I need to remove all this over here. Essentially, what I’m doing right here is I’m actually making it so this application will take live results from the website --- and we’re going to say the data provider is the --- what are we calling that --- directory list. Okay, no complaints yet hopefully. Nope. Okay, then the last thing I have to do is I need to bind it so it actually has live data. So instead of having just a demo text of my name, we’re actually going to have that as the --- well, the first line is going to be the common name that comes back from our
directory. The second line is going to have the department number, which if you work on this campus, department number actually stands for the department name for some odd reason. And I’m going to go and replace some other quick things on here as well. These are just quick programming things to actually make it so the demo works. Let’s go run this application right here. So we run this in the debugger. So notice now that when the application runs, it doesn’t have that list of me, you know, 15 times. What I did is I took that listing, that repeated list, and actually bound it to what’s coming back from the Internet. So let’s go actually search for something. I’m going to click in here, and let’s search for, let’s see here, let’s pick on Dave Mulder. Is he still in here? Oh, there he is. So, let’s say M-U-L-D-E-R. Actually, I forgot one thing. This is a live demo, so, you know, cut me a little bit of slack here, right? And I forgot to do --- I forgot to do what they call change handler, which actually executes that. So, I could have been typing all day long and it would have said, hey, I’m not going to do anything today. So we’re going to go back to that text input, and I’m going to say whenever that changes go and run my little function there to actually execute that change. So let’s go run it now. And let’s pick on Dave again. M-U-L-D-E-R.
Assuming that the Internet is working. There we go. So, here we go, David Lawrence Mulder, if I were to click on that, it actually shows information. Actually, right now it wouldn’t show that information because I haven’t hooked it up, you know, hooked that via code back up. So if I actually click on him, it would show me. I’m just that popular. But again, I can go and run this on the phone over here gain. So I can go run configuration, say deploy it to the phone, then pull the preview back up here, and while it’s sending it over to the phone --- it does take a few seconds to bring it to the phone. So it’s pulling the application back up. Let me go --- because I can’t really type in and have you guys see, I don’t think. Maybe. Of course, the WiFi dropped on this. So I won’t be able to show you the live results. Either way, you guys get the idea. It will actually execute the same on the emulator as it does on the phone.

As I’m running out of time, I’m going to end the live coding with that. I’ll finish up with the rest of my presentation. So key points. Flash Builder, everything you do in Flash Builder you can also do in Flash Professional, if you don’t have a coder available. So if you have somebody who is fairly familiar with using Flash Professional as a tool, they can deploy to these devices
exactly the same way. Because I’m using the beta software over here, actually the Flash Professional tool is a little further advanced as far as deploying to Android and deploying to the Apple operating system, or the Apple Suite of operating systems. They actually have a little box that says, yeah, deploy to this, deploy to this, and away you go. Now, these applications that I just created here, I can pretty much take those same applications, upload them to the various stores, to the Android Market and the Apple IOS Store. I can do that today. For the Apple Store I would actually have to purchase a certificate from Apple. For Google, there’s nothing you actually have to purchase. You can create your own certificate to upload there. I think the barrier to entry for the Android Store is $25.00, whereas Apple is, I think, $100.00 a year maybe, I don’t know, they always change that on me. Like I said, Flash Builder 4.0 is available today, and you can actually develop Android and IOS applications. There’s a few things you’re actually going to have to run command line. If you upgrade to this beta software that I just showed you just now, which, by the way, looks exactly the same as the Version 4 of the software, it just has that extra button on there that lets you deploy to the Android. That’s also available today as well, and includes the workflow that I
just showed. IOS deployment, you still need command line if you’re not on a Mac operating system. So that’s the only weird thing is, you know, Apple likes to do their Apple things --- yeah, I’ll leave it at that.

So, a lot of these things were actually already covered by some of the previous presentations, so I’m just going to run over them pretty quick. Mobile applications, you know, you really don’t want to make users think when they use it. They should be extremely intuitive, they should not have a whole lot on the screen. A., there’s not a whole lot of space on the screen, and B., if the user is presented with 15 different options and they’re driving down the road, you don’t want to cause that crash because you make them check ten boxes that they really don’t need to see today. Simplicity is really key. There’s not a lot of rooms --- okay. And also there’s not a whole lot of room to communicate with the user as well because you may be limited to, let’s say, 100 pixels wide. There’s not a whole lot of space to actually put things on the screen. So make things as simple as you can. You need to make your buttons and UI elements as big as you can because, again, people have fat fingers. My fingers are actually not so bad, but I know a lot of people have fingers that cover half that screen. You know, you make sure that you make it
intuitive enough and you make the buttons and the UI elements large enough so they can actually hit these things when they hit them, because unlike a mouse pointer, there’s no --- I mean, a mouse pointer, yes, you can make big, but it still has a little point at the end. A person’s finger, you have no idea how large that input is. Another thing is you need to make text big. 21-point font is the smallest you would ever want to go. If you actually download the directory application on your device that I just created over here, I actually used 21-point font, and I actually think that looks really small. Most of the time I would suggest you use 32-point font if you can actually do that. Don’t think in terms of clicks and drags. When you interact with devices people are expecting to interact with these devices a lot different than you’re used to on the desktop. Things like swiping, so like, you know, moving your finger across the screen. Pinch and zoom, so you actually pinch an image smaller or you zoom it in larger by creating it larger. Things like throws. That’s kind of popular with the IOS platform. You don’t see it a whole lot in the Android. Rotating the device, you know, so one of the interactions that you actually have is to rotate the device to something else so you can essentially make your application landscape. Another thing I thought was
interesting is some applications use shake, twist, twirl, yell. You can actually use those input types. So one neat thing about mobile devices, all of them have microphones. You can literally have it so one of your input methods is to yell at the thing, to scream at it. It’s possible. Always remember the network may not necessarily be available. Make sure your applications are aware of it, that they cash data, cash data very frequently, and these devices are getting larger and larger so you actually have more and more space to actually cash this data. If you’re working with the Apple system, they have built-in databases. On Android they have a different type of databases that you can actually store data in. Not a whole lot. I wouldn’t store gigabytes worth of data on it. But a lot of times you’re talking about searching results, you’re talking about frequently downloaded content. You can throw that on these devices. Test on every device you can get your hands on. For me, I’ve actually got access because Adobe loves me and all that fun stuff. They actually keep sending me devices as they start porting their error system over to these different devices. So I’m kind of blessed in the fact that I can go to Droid X, go to Droid 2, go to Samsung Vibrant, go to any of these other types of devices, and actually test my applications. My
suggestion for, if you don’t have hook-ups like that, is either get a big pocketbook and start buying these devices --- by the way, they don’t have to have service plans --- or become great friends with people who have these types of phones. So you can say, hey, can you go check out my app, or come over to my house, I’ll buy you a beer if you run this app on there so I can actually find out if it works or not. Find as many different varieties of phones as you can and start testing with them.

Some of the resources, if you want to play with the Adobe Flash Builder “Burrito,” Labs.Adobe.com. It’s a free download. I think it is a time bomb for a few months or something like that. But it’s definitely a beta so it’s not fully fleshed out but it works pretty well. I mean, you saw just a few points over here that I just had to point out, but otherwise it works really well. All the other applications I showed, the Device Central, Photo Shop, Catalyst, are all included Creative Suite CS 5. They’re included in the web versions and they’re also included in the Master Collection. For those of you who work at a university or some other educational institution --- great discounts. I can get the Master Collection for something like $400.00. I really didn’t touch a whole lot with HTML 5, but I’ve been playing with that as well to
make applications as well. One that just came out a little while ago is jQuery Mobile. Definitely take a look at that if you plan on making any HTML-based applications. What’s really neat about that is it talks about all these points we’ve heard in all the previous sessions about the size of the buttons, about detecting different screen sizes and things like that. It will take care of all that for you.

Finally, I want to talk about Adobe TV. TV.Adobe.com, they have hours and hours and hours of presentations on their website that show you how to use these applications, how to make mobile applications, how to make applications for even TV’s. They’re all available for free, very similar to production quality for like Lynda.com or things like that. In fact, some of the people who actually work for Lynda make videos for them as well. So definitely check that out. They’ve got hundreds of hours of content on there as well.

Other than that, that’s pretty much what I have. Are there any questions at all about what I presented or showed? Yes.

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

MR. KWIATKOWSKI: The question was binding the label with the input box. I actually created an input box which
was a shape and then I went into Text Tool and I just laid that over as a separate layer. Within Catalyst I said that input --- or the text was as part of that input box. So because I selected them both and I said, this is an input box, Catalyst knows that, hey, this is text, this is a shape, you’re going to use those together in that way.

MEMBER OF THE AUDIENCE: That happens with Catalyst after ---

MR. KWIAKTOWSKI: That happens within Catalyst, yes. Yes?

MEMBER OF THE AUDIENCE: What’s your blog ---

MR. KWIAKTOWSKI: Oh, yeah, my blog. My blog is Quetwo.com. I should probably put that on the screen here because like my last name, it’s not easy to spell. I’ve got a whole bunch of things not necessarily about mobile development. I’ll have more on there pretty soon. Let’s see here. So everybody can actually see it. Maybe that’s too big. Oh, who cares. There we go. So, that’s my blog right there. Again, I’ve got a bunch of different stuff. In fact, last week I posted how to make it quick television application because I’m actually building applications for my Samsung TV right now, which is pretty cool. It brings a whole other level of web applications and actually developing richer Internet applications because now, like I
said, I can’t actually touch my TV. I actually have to use the remote to do everything. There’s no keyboard either.

You had a question back there.

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

MR. KWIATKOWSKI: So the question is, when I ran the device --- or when I ran the application on my phone, did it install new application. Yes, it did. It will, if it sees that same application on there, because of that application I.D., it will remove previous versions every time I run it. But it does actually live on the phone. If you guys want to come up here and actually play with the devices, I’ve got --- I’ll deploy it on multiple devices here so you can actually play with it as well. Feel free to play with these phones. But, yes, it does actually replace them every single time I iterate that.

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

MR. KWIATKOWSKI: The question is every time I run it, you know, is it possible if stale data or data over there causing issues. I have seen that in the past primarily if I’m storing data within the application scope or things like that. When you deploy applications to the phone, one of the options here is to clear application data on each
launch and that will actually wipe out that user space for that particular application. So that will clear all that out. So what that actually does is it does a full uninstall, it clears that memory, and then it reinstalls it. It takes a little bit longer but it does fix that.

Any other questions? Yes, Dave.

MEMBER OF THE AUDIENCE: What kind of situation would a developer not want to use this technique?

MR. KWIATKOWSKI: Okay, the question is when would a developer not want to use this technique for creating an application. There’s actually quite a few scenarios. What I created right here is going to be pretty much restricted to one --- not this application in particular, but if you go too crazy with the design, you may run into situations where if you want to deploy that same design to multiple phones or things like that, or multiple screen resolutions, you’re going to start running into issues because the screen resolutions on different phones may differ. There’s other techniques you can use to correct that and usually what I would do is I’d start with Photo Shop, kind of split things up, and then bring in Catalyst and then kind of line them up, so I can say always stretch this image this way, or things like that. So this workflow is perfectly valid. It will kind of limit your ability to make quick
applications over multiple different types of phones. Now, one thing I do need to say with this as well, is again, accessibility wasn’t in mind when I kind of wrote this really quickly. If you were to write from the developer aspect, the developer point of view within Flash Professional or within Flash Builder, the accessibility is forefront, it’s turned on by default, and then you can go that way. Again, you can add the accessibility all in here. It’s actually really easy to do. It’s just that it takes an extra step if you’re starting from Photo Shop going back.

There was a question over here. Yes?

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

MR. KWIATKOWSKI: This application I would expect to be about 1 MB or so because I’m using all graphics. That is fairly large in the world of applications as far as iPhone and Android applications, considering it doesn’t do a whole lot. But, again, we have to realize that we started with a Photo Shop file and went that way. I could have done a lot of different things within Catalyst to optimize those images. I could have made it so the background wasn’t in gradient and things like that, and it would definitely cut down on the size that’s required on
it. So, we’re not talking about a multiple megabyte file, we’re talking about one megabyte, which is still pretty acceptable for most people.

Yes?

MEMBER OF THE AUDIENCE: How much effort do you think it requires to turn into production code if you don’t have Adobe (inaudible) platform.

MR. KWIATKOWSKI: So, as far as using --- the question was how much effort is there to be had if we were to convert the same thing to a non-Adobe air application. Um, probably quite a bit. You can use Catalyst and you can use all that to prototype your applications, but then your developer is going to have to start from scratch if they’re going to use anything else. The same technique also could be used to create a Flash application that would run within the browser with not a whole lot of work. Actually, when I exported from Catalyst to bring it into Flash Builder, that’s what it thought I wanted to do. That’s why I had to do the copy and paste over there. But as far as using anything other than Adobe Air, that depends on what you’re targeting. Now, one thing to note with Adobe Air is it is actually creating native applications for anything other than the desktop. So what’s actually installed on the Motorola devices there is an actual native application that
runs on those. And that’s why when I deployed to IOS it’s also a native application as well.

Yes?

MEMBER OF THE AUDIENCE: Given it’s a native application, there’s obviously an Air run timepiece that’s going to these devices. (Inaudible) or with every app?

MR. KWIAKTOWSKI: The Air run time, on Android devices, on Rim devices, on Nokia devices, is a separate run time. For the Apple, because of the restrictions that they have for devices within the market, they actually include it on there so you always have it. So devices --- so when you deploy to an IOS device you’re going to see that the application size is about 2 MB or so because it has to have that 1 MB run time with it.

Yes, in the back.

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

MR. KWIAKTOWSKI: The question was Apple hasn’t always been friendly to developers who develop on things other than Apple toys, let’s put it that way. Right now you’re allowed to do this. Apple, in, I believe it was the beginning of September, made a change in their market agreement that allows you to write applications and deploy applications on things other than X Code. That was a
change from, I believe it was August, middle of August --- no, middle of April, sorry --- where they started to
disallow that. Now, their decision to disallow that also
disallowed quite a few other types of application types,
including some of the most popular games that were being
deployed at the time, including those written in Away3D,
which is similar to this.

I think we’re just about out of time for questions here. If you have any other questions, feel free to find me. I’ll be sitting up front over here. If you want to
play with the applications, please do. They’re sitting up here. Otherwise, thank you very much.