.NET Rocks!
The Internet Audio Talk Show for .NET Developers
With Carl Franklin and Richard Campbell
msdn
http://www.dotnetrocks.com

http://www.dotnetrocks.com

Carl Franklin and Richard Campbell interview experts to bring you insights into .NET technology and the state of software development. More than just a dry interview show, we have fun! Original Music! Prizes! Check out what you've been missing!

Text Transcript of Show # 234
(Transcription services provided by PWOP Productions)

Frank Savage on Programming the XBox 360
May 3, 2007

Our Sponsors

Geoff Maciolek: The opinions and viewpoints expressed in .NET Rocks! are not necessarily those of its sponsors or of Microsoft Corporation, its partners or employees. .NET Rocks! is a production of FranklinsNet, which is solely responsible for its content. FranklinsNet - 'Training Developers to Work Smarter.'

(Music)

Lawrence Ryan: Hey, Rock heads! Stop using your daughter's DDR Mat to play Strike Commander and listen up, it's time for another stellar episode of .NET Rocks!, the Internet audio talk show for .NET developers with Carl Franklin and Richard Campbell. This is Lawrence Ryan announcing show #234 with guest Frank Savage, recorded live Wednesday, April 25, 2007. .NET Rocks! is brought to you by FranklinsNet - 'Training Developers to Work Smarter', and now bringing the Just-in-time Team System Class with Joel Semeniuk onsite for your development team, online at www.franklins.net. Support is also provided by 'Telerik' - combining the best in Windows Forms and ASP.NET controls with first-class customer service, online at www.telerik.com; and by CoDe Magazine - the leading independent magazine for .NET developers, online at www.code-magazine.com. And now, the only man I know with an anatomically correct me for the Wii, Carl Franklin.

Carl Franklin: Thank you very much and welcome back to .NET Rocks! This is the second show of the week of MIX 07. It's MIX week and we're still not there, right Richard?

Richard Campbell: We still aren't there. But that's okay;

Carl Franklin: but I heard it was good.

Richard Campbell: I'm sure it is.

Carl Franklin: Yeah. Great show we did with Brad Abrams.

Richard Campbell: Had a lot of fun.

Carl Franklin: Boy! There is so much new, great stuff coming up; there's so much stuff we didn't even talk about with Brad.

Richard Campbell: Well, and MIX was just one big stream of announcements; so many things going on in the Web space especially.

Carl Franklin: Well, I can't wait to dive into some of that stuff in more detail on this show, on Hanselminutes and on dnrTV. And who knows, you would even touch on some of it on RunAs but…

Richard Campbell: You better believe it.

Carl Franklin: Alright, well let's read the emails. This one is from John S. Brown, and the subject is "Mumbles and the other guy." Okay; sounds like a movie, you know?

Richard Campbell: Yeah.

Carl Franklin: "Dear mumbles and the other guy, this is a rambling diatribe of life, liberty and the inevitable divorce."

Richard Campbell: Nice.

Carl Franklin: "Six to nine months ago, the radio in my car died; since my car already had a lot of miles on it - 228,000, I reasoned that I would get a new car, so, no point in replacing the radio. As you can quickly surmise, no new car, no new radio and still an hour long commute; one can only ponder their own thoughts for so long, so I stole my wife's MP3 player that I gifted for her for her birthday about a month ago and downloaded my first podcast ever. And yes, I got lucky and downloaded the .NET Rocks! interview with Scott Ambler on Agile, episode 226. I have been a podcast junkie ever since so much so that my wife is demanding her MP3 player back and is threatening to divorce if I don't "take those damn headphones out of your ears". Today I felt compelled to write in, because the current episode with Jeff Atwood was positively painful, not because it wasn't a great show - it was - because today I was being chastised for "asking for ridiculous hardware for my developers". I am a project manager and a software architect with one of the beltway bandits - a.k.a. federal contractors, and working on a four-year software development project for a government agency. This is a system that integrates fifteen separate databases, two web servers, 100 plus data entry pages, 50 plus reports, 3000 users and more than 500 gigabytes of data. It uses a WinForm client with Infragistics and VSTO to build 20 plus documents, .NET Remoting, Web Services, Crystal Reports and 4000 plus source code files, which are compiled many, many times a day. We do all of this on crappy, 2.8 GHz Dell's with only 1.5 Gigs RAM and a mixture of 20" and 17" monitors. These are three-year-old machines, and I was only asking for an interim solution of 24" monitors, and upgrades to 4 Gigs of memory for a whole six developers.

(00:05:16)

You thought I'd ask for Limos to pick us up for work in the morning. See? I warned against this. But yes, it is the government, and they can only buy equipment for the whole agency at one time.

Yes, the secretary surfing the Web has that same hardware configuration as 90% of the software development teams. We have not enjoyed any item of Jeff Atwood's Bill of Rights, so, why don't we buy our own? The government agency forbids us from using anything but their equipment. They'll spend thousands and thousands of dollars for overtime, but heaven forbid, they make a small capital purchase for productivity." Dude, I'm so sorry, it sucks. And this, in the city that houses the actual Bill of Rights; it just doesn't seem fair. He goes on, "So, why did I call you mumbles? For the first 15 to 20 episodes that I listened, I kept hearing you advertise these jobs with Infusion at shrinksfer.com or was it shrinkthefar -- well anyway, I didn't know what Greg Brill did at Infusion, but I assumed it was for the American Psychiatric Association. It wasn't until recently where the other guy Richard, spoke clearly enough to understand shrinkster.com. Anyway, Carl and Richard, keep up the good work; I'm going to go cry in my beer over monitors and memory. P.S. How about some swag to soothe a guy's feelings for when I tell them? I went to the boss to get you new hardware, and all I came back with was this lousy DNR Mug." We're going to add the DNR mug into Jeff Atwood's Bill of Rights, that's what we got to do.

Richard Campbell: Absolutely.

Carl Franklin: Well, he is John from Washington D.C. John, you bet, man; I'll send each one of your six developers a DNR Mug.

Richard Campbell: There you go. Go to give him some love somehow.

Carl Franklin: Absolutely.

Richard Campbell: That was a great email.

Carl Franklin: It was; I really love it when people who can write send me emails.

Richard Campbell: Go off on a little tangent there.

Carl Franklin: And we've had a couple of emails about shrinkster.com.

Richard Campbell: Yeah. It's just like the other URL shrinkers, it's just, it's written with .NET, so we like it.

Carl Franklin: That's why we like it.

Richard Campbell: Yeah. I have an email as well. "Richard and Carl, just listening to the DNR show on Reporting, where you were talking about using a SQL Server Mirror to run your reports off, the only problem with Mirrors, is that the Mirror is actually quite busy running all the same transactions as the Principal, and thus is constantly restoring state and is not available, or online, for you to run reports off of. I guess you could use things like database snapshots to take an instant copy and run reports from that, which would probably be quicker than doing backup restores. I'm pulling into Waterloo now, so take care, and thanks for all the excellent shows coming out of Pwop!. Best Regards, James Saul. And it's Waterloo in the U.K. as opposed to the University of Waterloo at Canada." And actually, James is completely right, "If you are actually using true Database Mirroring, you can't do anything with the server that you're Mirroring to -- that's a failover strategy, so that's really quite limited. We were talking about techniques for making copies of data at intervals, whether that be log transfers or backup or storage cycles, lot's of different ways to go about it." But, thanks for clarifying that; don't use database mirroring to report off of it - doesn't work.

Carl Franklin: And see what you get for tuning in this morning? Excellent. So, we got some code camps to announce here, the Ann Arbor Day of .NET, May 5th, that's this weekend, this Saturday I believe at shrinkster.com/cuk.

Richard Campbell: Also this weekend, May 5th, the Austin Code Camp at shrinkster.com/o9e.

Carl Franklin: Also, the West Michigan Day of .NET is going to be May 19th. You can read about that at shrinkster.com/n1h.

Richard Campbell: And the Philly .NET Code Camp at www.shrinkster.com/0i7.

Carl Franklin: Yeah. And finally, the Raleigh North Carolina code camp Trinug as they call it -- I don't even know - whatever -- you don't have to figure it out - go to www.shrinkster.com/o17 - and that's June 23rd. Now, we have DevTeach coming up here; May 14th through 18th in Montreal.

Richard Campbell: Montreal Quebec. We will be there.

Carl Franklin: We will be doing .NET Rocks! – we'll be doing a panel - some sessions, and I hear Roy Osherove is going to bring his guitar and sing us some songs.

(00:10:06)

Richard Campbell: There is going to be some jamming going on, don't you hold back, Mr. Franklin, I know you will have your guitar as well.

Carl Franklin: Maybe, maybe I don't know. I like listening to Roy, he is more clever than I am.

Richard Campbell: Yeah, he is got some clever lyrics going on.

Carl Franklin: Yeah, he is good.

Richard Campbell: He is the Israeli Ted Patterson.

Carl Franklin: No, no, no he is the .NET, Adam Sandler, is what he is.

Richard Campbell: He certainly looks the part anyway, and the other conference in mind TechEd.

Carl Franklin: Right.

Richard Campbell: TechEd US Orlando, Florida, June 4th to 8th

Carl Franklin: Which may or may not be sold out so if you are still on the fence, you probably want to go.

Richard Campbell: Yeah go quick, we'll be there.

Carl Franklin: Also the New York tour, Greg Brill's Infusion, New York City offer, at shrinkster.com/kh6. If you want to be teleported to New York City and get a New York City salary for a year and live rent-free in the city, check it out, shrinkster.com/kh6 but you got to be badass to apply. Also for badasses in the ASP.NET area, if you are willing to be relocated to Washington DC or if you are in the area go to shrinkster.com/mmj . There is a killer opportunity for you down there as well. All right Richard, let's bring on Frank Savage. Frank started in the Game Development Business at Origin Systems in 1991. First game was Strike Commander, then was the lead on Wing Commander III. He left Origin and went to work for FASA Interactive, or FASA, in Chicago and did the first MechCommander Game, there. FASA was acquired by Microsoft, in 1999 and he finished MechCommander 2 at Microsoft in June 2001. At that point he left and went to work for Xbox just before the first Xbox launch. He worked for the Advanced Technology Group which helped game developers get the most out of the Xbox console. While working there, he began the first discussions around XNA and he was the third member of the dedicated XNA team as the development manager. And they've shipped their first release and are currently readying the second release for shipment even as we speak. Will you please welcome Frank Savage, superstar! How are you doing Frank?

Frank Savage: Good, how are you guys?

Carl Franklin: You probably realize how many fans you have, but you probably try to keep a low profile, don't you?

Frank Savage: Yeah, I hear every now and then, in the forums, someone says, oh it's that Frank Savage and then points out that I worked on Wing Commander III, or Strike Commander or one of the MechCommander games. They remember me from having worked in ATG, I worked with a lot of game development companies. While in the advanced technology group, on Xbox, helping them get their games up to speed, finding performance issues, doing all that kind of work to really help them, get the most that they could out of a console and really get the performance that the console was capable of in the last generation. Didn't really do that a lot with the current generation, but I was pretty deeply involved in the hardware design for the current generation, which was fun as well, so.

Carl Franklin: I got to imagine that being you must be like, trying to swat flies all day long. Hey, Frank, tell me how to get the cheap code in Wing Commander2, hey Frank. You must be conferences or whatever you probably have swarms of geeks that just, want to know, is that accurate.

Frank Savage: Yeah, that and my parents keep calling me for Tech support and all that as well too, which didn't help either so.

Richard Campbell: We are all up against that.

Frank Savage: Yeah, everybody.

Carl Franklin: Does your mother play Wing Commander?

Frank Savage: My mom, is a huge gamer actually, which is surprising, and I think, both to her and me. She started out very much on PlayStation 1 and played a lot of games on PlayStation 1, Xbox came out, she got an Xbox, was a very early adopter of that, in fact would, well, even before that when Wing Commander III shipped, my mom would go the CompUSA's and stores like that and say that's my son and show that thing. So, they would give her all of the posters and all of the marketing stuff that we would send them. So, my mom at one point had a shrine to me in the bedroom, with all this Wing Commander III, MechCommander and Strike Commander stuff that she had accumulated and my Origin Most Valuable Employee Plaque was in the shrine, and the whole nine yards. So, it's a little bit harder now, well it was a little bit harder until the XNA, stuff started to really kick in and because in the events technology group, we were

helping a lot of games but I wasn't showing up in the credits for a lot of games and I couldn't tell her what games were coming out that were really cool, but didn't really have that kind of exposure that I used to have.

(00:15:01)

Carl Franklin: So, have you come to terms with the moral implications of your job, which is that you are reducing the gene pool of geeks everywhere by keeping them glued to their computers, instead of going out to meet girls.

Frank Savage: Yeah, we are pretty active at that. But the cool part of this is that I am actually actively training my kids to be geeks, and they are making a lot of progress. My daughter is three-and-a-half, runs around doing Star Wars quotes and playing with her little Star Wars action figures and my son, who's just turned seven years old, has a couple of level 20 and level 30 characters in the World of WorldCraft already.

Richard Campbell: Oh, man!

Frank Savage: He's very nearly finished commanding Conquer 3, so he is pretty hardcore for a seven-year-old. He has his own laptop with the high-end video card in it, because he kept playing with mine and I can't get any work done at home. So, they are probably going to be where the Savage line ends because they are never going to get away from the computer.

Carl Franklin: That's what I'm thinking.

Frank Savage: But I have done at least what I could for this generation.

Carl Franklin: Good, good, good. Now, when you think of gaming, you don't usually think right after that, .NET?

Frank Savage: No, it's generally we found that, that wasn't the case but about two years ago, this started to change a little bit, when I was in the Advanced Technology Group, actually three years ago now, one of the things we found game developers starting to do, was we had just finished a thing called Managed DirectX, and Managed DirectX, was done by a single developer called Tom Miller, who actually now works on the XNA team.

Richard Campbell: Wow, one guy did Managed DirectX?

Frank Savage: Yes, one guy did Managed DirectX.

Carl Franklin: Now that man has no life.

Frank Savage: We made sure that he had no life for the first release of our product as well, which he is eternally thankful to me for, but the idea was with Managed DirectX, you could actually go and write some really cool, very fast duration Windows Tools, to do your game developments. You couldn't run .NET on Xbox or Xbox 360, but you could run it on Windows and you could build your tools for your game using Managed DirectX and using Windows Forms and things like that. You can get tools up and running very, very quickly. So, we began to evangelize this to game developers, and today, it's hard to find unmanaged tools in game development houses. Very nearly everything that we encounter is very .NETty in terms of how they built the Windows Interfaces and the tools and Date Editors and Level Editors and things like that for their game applications, is very, very C#, .NET oriented today.

Richard Campbell: And maybe we need to paint a better picture for folks who've never worked in the game industry about how game development is really been done up 'til now.

Frank Savage: Sure.

Richard Campbell: Sort of in house, build your own tools, everybody is a C++ programmer kind of mentality.

Frank Savage: Exactly, so back in the day, it's hard to believe that I am saying that these days, but it's been 16 years since I started doing this. And 16 years ago, we were DOS with 16-bit Borland compilers. Games were extremely difficult to debug because you had different segments and offsets that are showing the same addresses is very easy to have. Uninitialized pointers that randomly pointed a different area in memory, things were harder by a large margin than they are today, and the tools were incredibly primitive. So, at that time even though, we starting to do 3D based applications like Strike Commander and Wing Commander III, the tools for these were very much at that point CAD-based programs and they were designed to do very high polygon, very detailed computer-rated design drawings, rather than very, very low polygon, barely textured 3D meshes that could be rendered in the technology at that time, because there were no 3D accelerators at that time. So, just getting those tools out with something that the game could use is nearly impossible and we wrote a lot of those tools ourselves, we had a 3D editor, called EOR that we wrote for Strike Commander and for Wing Commander III and that tool required a very different mindset than the 3D modeler has today and because it was designed by programmers for

programmers to make 3D art not a 3D artist to make 3D art, and it became very difficult to really understand what those guys needed, because they themselves were so new to the business that, it was hard to even figure out, what would help them, and what wouldn't. As we moved on the MechCommander style games, the tools got better and better and more and more of the 3D packages began to understand that these people needed low polygon models. But okay, great now, I have got the low polygon model, I still now have to build my terrain, have to place all the buildings or I have to lay out all the corridors, I have to put the traps in the place, I have to the put the monsters down, I have got put some AI around the monsters. The doors have to open when I push the switch, the race cars all have to stay glued to the track, I have got to get real physics on to the tires, I got to put all the car data in, all the aerodynamic data, the track characteristics, what if the track characteristics when they're wet.

(00:20:12)

There is an enormous amount of data, associated with games and the data entry for that used to be very crack open notepad and start typing. As time went on, the tools got more and more sophisticated to the point now, where there are games, that actively SQL back ends to store all the data, that's going to go into the game and then the SQL, there is actually a report generator that runs on the SQL database that actually steps out of the format of the game files, so that the games is going to actually be able to run.

I have even heard, tell that there are games that have actual SQL embedded in them, in order to be able to just create the database themselves and get the data out, because it's so big and so, hard to figure out these days. So, games have gotten quite a bit bigger over time as well, when we started in the business.

Carl Franklin: In that process, in their earliest days, where you not only tech guys, but were you coming up with the stories and all the art work and all that stuff too, or has it always been a collaboration between the creative people and the developers?

Frank Savage: So, when I first started in the business, it was just starting to differentiate, where you were starting to see -- the programmers used to do everything and in fact games, older games had what was called, what's still called today programmer art in them and you'd look at them and go, wow, a programmer did that, because it was not so good. And the design of the game was very much a programmer centric thing as well. Again, the programmers tended to be gamers so, we knew what we liked to play in a game but sometimes it was hard to nap that to a very large audience for example and so a game that had a very strong appeal to the programming staff that worked on it, may not necessarily be a game that had a very strong appeal to a wide audience. So, as time went on, the roles began to differentiate somewhat, and there was a programming staff that was responsible for the implementation of the code. They became a design staff and the design staff was very much in charge of everything from the lowest level data entry, all the way up through the high level game design features like how the AI should behave, what the story was going to be, what did we want the player to feel at this point in the game, what was the flow of the story, does this happen, then this happened, how linear was it, how non-linear was it, how where could they go off into the weeds and explore for a bit and then come back to the main plot and stuff like that. So, all of that kind of stuff, it took a while for that to evolve into the design side and then post that year then it also became a whole bunch of dedicated artists and the dedicated artists started out as very 2D artists because the games were still very two dimensional even for the scripted portions of them, like when we are telling the story in Strike Commander, it was all 2D art that was developed by 2D artists but then 3D artists became necessary as the games moved into polygonal models that were texture mapped and 3D hardware began to become more prevalent. The game got more and more art intensive on both the 2D and the 3D side. So, today, a typical game house has a couple of hundred artists working on the game, anywhere from 5 to 25 programmers working on it and a design staff that is directly proportional to the amount of content that can actually be in the game and that can range anywhere from a dozen to again a couple of hundreds like the artists, so you are talking about something near the scale of like the World of Warcraft, so.

Carl Franklin: Now wasn't Wing Commander an Electronic Arts game at one time?

Frank Savage: Yes, so Origin Systems was originally an independent game publishing company and they were acquired by Electronic Arts, in 1992.

Carl Franklin: Okay, all right and so when did FASA come into that?

Frank Savage: So, when I finished Wing Commander III, I was actively looking for something else to do, Chris Roberts's contract with Origin Systems was going to end in about a year with EA and he was going to go off and do

his own thing, he wanted me to come with him and I was like, well, I have just finished this Wing Commander, thing and kind of want to work on games more and you kind of want to go off and do Hollywood things and I am not sure that I want to do that. So, I left Origin and went to FASA Interactive Technologies up in Chicago, and I was the 8th employee there, I was also the first technical employee, so I was with one leg in the networking and emails and Windows Servers and things like that, which was a very different role than making games.

Carl Franklin: What confuse me was MechCommander and Wing Commander even though they have commander in the name, there is no relationship there company-wise, right?

(00:24:50)

Frank Savage: Correct, yes apparently, I am doomed to work on commander titles, for the rest of my career. So, one of the reasons that I got out the game business, was I feel what I needed to say in the Commander genre and I really want to kind of move on to some else and…

Carl Franklin: Oh, that's funny. You've made your piece.

Frank Savage: That's right you know and I guess, I said what I needed to say.

Carl Franklin: I love it.

Frank Savage: I have explored it.

Carl Franklin: All right let's fast forward to XNA, how did that all come about and how does it relate to .NET?

Frank Savage: So, when we started in XNA at Microsoft, now this was back in 2004, when J first announced it at Game Developer's Conference…

Carl Franklin: J?

Frank Savage: J Allard, who was head of the Xbox division actually. He had this vision for let's make game development easy, or easier at the very least. And what could we do to help core game developers, people like EA, Activision, UbiSoft, people like that, to get their games faster, easier onto the platforms that we had and we realize that, while, you can make the hardware as easy to work with as you possibly can, it's all about the software, it's all about the tools and the compilers and being able to connect and debug and get profiling information and that kind of stuff readily from the console, otherwise it's extremely difficult to get any kind of good game done on it. So, initially our task was trying to figure out how do we help these guys and what technologies did Microsoft have to bring to bear to be able to do that. As we began investigating that, we actually did a bunch of work on build systems and we did our first kind of hobbyist and enthusiast thing and that we actually bundled up MechCommander 2 all the source, all the content and posted it as a shared source release and you can still download that from Microsoft today and it's about a gigabyte in size, not small by any stretch of the imagination, but it is the entire game and you can compile the entire game out with Visual Studio 2005, using the 8.0 version of C++ compilers, we moved it all, modernized it all to that.

Carl Franklin: No, I am sorry this is MechCommander you said, right?

Frank Savage: Right, this is MechCommander 2 that we did here at Microsoft. So, when we released that we found that there was this enormous hobbyist and enthusiast community that was very excited about being able to get their hands on a game but the number one piece of feedback, we got from them was cool, but this is real big. I just want to do something kind of small and you've given me, you've given me the Titanic and I need a robot, and so we heard that loud and clear from the customer base, while we were hearing that from the customer base, the .NET Compact Framework Team, who at that time was being lead by Mike Zintel, came to us and said, hey wouldn't it be great if the compact framework ran on Xbox 360? And we said, yeah, that would be really nice. Why are you asking us that? And they said, well because we would like to divert some resources to doing that, is there anyway we could get some Alpha Kits or whatever, and I said, you know what, I will drive them to your building. So, that was in August, 2005, by November 2005, they actually had it working on the Xbox 360 Alpha Kits, and basically…

Carl Franklin: When you say, it.

Frank Savage: So, we had .NET framework, a subset of the .NET framework tough Managed DirectX, ported to work on both, on the .NET Compact Framework and the Desktop CLR, running on Windows, a Pocket PC and the Xbox 360 with almost exactly the same code.

Carl Franklin: Wow.

Richard Campbell: It's pretty exciting.

Frank Savage: This excited a lot of people over here as well, and we went huh, maybe we are going the wrong way on this. Maybe there is something to this whole hobbyist, enthusiast,

community thing and maybe we really can't crack it open. Because the biggest hurdle we had, which was this technology, seems to be well on its way to being solved. So, we went back and did a prototype on the actual release, Xbox 360 hardware and that was significantly more challenging than the Alpha Kits because the release hardware had a very tight security model and this is a console that we don't people going in and hacking and stealing games and running their own stuff on, without going through our certification process. So, we had to make sure that in putting the .NET Compact Framework on it that we didn't inadvertently, make the Box really easy to hack and you can run anything you want on it, so forth.

Carl Franklin: Yeah, that was first concern, right, is that people stop buying PC's and start buying the low cost Xbox 360's.

Frank Savage: Exactly, so we made sure that the security model was such that the interface, if you will to the Xbox title libraries, or what the Xbox hardware can actually do, was very, very, tightly controlled by us and by the .NET Compact Framework team. So, for example today, you can't actually get to the networking at all, there is no multiplayer connectivity right now, and that's coming actually in the fall, but there isn't any right now.

(00:30:00)

So, we did that again on purpose, because we wanted to make sure that we open it up in a way, that supports the live business model, you know, meets all he business and political and socio criteria that exist here, technically it's not hard to do, it's just making sure that we don't ruin people business models.

Richard Campbell: Doing it in a way, that doesn't actually break live or introduce a new class of virus or there is all kinds of things that go terribly wrong with that.

Frank Savage: Yup, exactly. So, they have the user mode Sandbox pretty much up and running, so that that CPU that's in the Xbox 360, there is actually three CPU's, there's three power PC cores, each one is running at three gigahertz. There is two hardware threads on each core. All of those can run in a couple of different modes, the mode that the game runs in a is essentially kernel mode, it's the boss, it can do anything it wants. We keep the games from doing anything malicious to other games or to live or anything else by going to through our certification process, it's a large part of what that does, but again if we are opening it, it's up to the community, if a bunch of community people can go and create games, using the .NET framework they can't really -- we want to make sure they can't inadvertently, or advertently do that to people. So, we created the user mode Sandbox, and again we had a very clearly defined way of getting out of. Part of the charter was also lets find a way to make an API, that isn't Managed DirectX, which was actually at the end of the day a pretty thin rapper over the existing DirectX functionality in managed code. But let's make an API that's really cross-platform, that had a goal of being 95% compatible between Windows and Xbox, so there was a very high probability that your game would recompile and just run on Xbox 360 after you'd gotten it working on Windows or vice versa.

So, we started off down that line. We also decided look, lets make a framework that's extremely .NET oriented, that smells and feels like the rest of the .NET frameworks do, and lets make sure that we target it for the right audience and the right audience for the first release was the very serious, hobbyist and enthusiast and also the academics and students. So, we've actually had a fair amount of adoption from universities, I'd like to talk about that a little bit later on, but there is a ton of universities already using this stuff, in fact they were using our beta bits, that's how eager they were to get their hands on it. So, we wrote -- so the framework component was being built while we were actually finishing out the .NET components and this is the thing to keep in mind and the Herculean effort that went on with an incredibly small number of people, that shows you what you can do, when you stand on the shoulders of giants literally, is in March of last year, we had the prototypes up and running and that was it, and we went back and not from scratch but standing on all of this technology that we had lying around, we were able to go from March to December and ship a working version both Windows and Xbox 360, that allows you to run games on your retail Xbox 360, you don't need any special hardware, you can go to the store and buy one today and connect it to your PC, your PC can upload your games to your Xbox 360, it can play them, you can debug on your PC's using C# Express, all of that is fully functional and available, and it's been available since December.

Carl Franklin: Now is that the XNA Game Studio Express, or is that something else?

Frank Savage: Yes. So, that's the Game Studio Express piece and to run that you need to C# Express first and then we install into C# Express and extend it.

Carl Franklin: So, now why only C#? Just curious?

Frank Savage: C# was the most obvious choice for our customer base, we were talking about game developer hobbyist and enthusiasts, which were typically, shallow C++ users. So, it seemed to map to the C# universe more easily than things like Visual Basic or any of the other .NET languages that exist today.

Carl Franklin: So, but it's Visual Studio, couldn't – aren't we talking about API's aren't we talking about a framework or are you talking about a boatload of source code.

Frank Savage: So, there was actually -- Visual Basic actually isn't quite, as .NETTY as it should be for example, requires other DLL's to be present from the old VB run days.

Carl Franklin: Like what?

Frank Savage: Actually, yeah so there is no Visual Basics support on the compact framework for example.

Carl Franklin: Wait a minute now, so what exactly do you mean by that? I can write programs in VB NET for the compact framework.

Richard Campbell: I can do that.

Frank Savage: But the compact framework itself doesn't support Visual Basic as it exists today in the 2.0 iteration.

Carl Franklin: Are you talking about like writing code on my PDA?

Frank Savage: Yeah!

Carl Franklin: Oh, well who would want to do that? I mean I want to write code for the PDA but I don't want to write code on my PDA.

Frank Savage: Well, that's the problem right, is that we want to run the code, on the Xbox 360 and that required DLLs that just weren't present.

Carl Franklin: Do you mean, you want to run Visual Studio on the Xbox 360?

Frank Savage: No, the VB run OCX, DLL and stuff like that has to be present on the operating system and they are on the PDA's but they are not on the Xbox 360 and we didn't have anyone to port those to power PC …

(00:35:10)

Carl Franklin: I understand.

Frank Savage: ... to make that go. So, Visual Basic got less interesting from that perspective as well and that there was a lot more technical work to do and again we were under a very constrained time line, we wanted to ship last December if at all possible and we had the Tech for C# pretty much done. And again it mapped to the customer a little bit better as well.

Richard Campbell: Now, it's pretty impressive to think this is all Express, so if you want to develop on your Xbox, you don't need to buy anything, accept an Xbox.

Frank Savage: That's correct, and the subscription, so that's the other copy out.

Carl Franklin: So, what kind of stuff can you do and what kind of stuff can't you do with the Game Studio Express?

Frank Savage: So, the Game Studio Express today, the limitations are basically mostly around the networking code. So, we have API's that do all the graphic stuff, the graphics API's are identical between Windows and Xbox. We have input API's that are identical between Windows and Xbox, you can plug in Xbox 360 controllers USB into your Windows PC. It will recognize and download the driver and that controller will behave and act exactly like it does when it's plugged in the Xbox 360.

Carl Franklin: You know, in comparisons of Web Development components come into the play, vendors start tossing in clichés like complete tool set of controls, superior performance, empowering users and hosts of others buzz words, but at the end of the day, what matters most to you, the developer. For our friends at Telerik, the answer boils down to simply getting your job done. Like saving precious time by customizing stubborn controls at design time or skinning new applications in no time. And how about no browser compatibility issues, that's a big one. Take the Telerik AJAX offering for example, the product was designed to quickly get you up and running with this new, yet complex technology and it just works. Forget about writing tricky JavaScript, forget about making end-to-end modifications to your application, what's best is that you can count on a wide range of resources, Sample Apps, Tutorials, Active Forms and of course Telerik's renowned support team. After all there is a reason why 89 % of Telerik's customers choose to renew their subscriptions. Experience that for yourself by testing a trial version of the most reliable UI suite for ASP.NET at www.telerik.com.

So, basically you can take a C# DirectX program and port it to the Xbox, is that what you are telling me?

Frank Savage: No, right now, if you use the XNA framework's that are in the Game Studio Express, they will simply recompile and run on the Xbox.

Carl Franklin: Okay, so the XNA Game Studio Express API's are kind of like using the DirectX API's is this what you telling me? They are sort of similar.

Frank Savage: They are sort of similar but again they have been somewhat simplified for our target audience and we've removed or at least hidden away somewhat the functionality that is confusing or hard for people to really rap their heads around. For example, creating a device in Windows, is actually a fairly long and drawn out process today, we'll have to get at the 3D hardware and its magic in our universe. It simply creates the device and maps it to the current resolution of your screen and sets everything up and you are ready to go and you can just start calling update and render.

Carl Franklin: So, you took it up a level basically, you're a couple levels.

Frank Savage: Exactly, yup and we wanted people to not have to worry about and again one of the biggest problems on the Windows side, is that there are scads and scads of different kinds of hardware configurations. We've abstracted away, a lot of that, as well, so that again you don't really need to worry about the hardware configuration, what you need to worry is that, you've -- the only thing you really need to worry about today, is the Shader Model implementation on the video cards, so we only support 3D cards and we only support 3D cards that have at least Shader Model 1.1 on them.

Carl Franklin: Now what is that? What is Shader Model?

Frank Savage: Video cards, 3D video cards today, are actually programmable and they're programmable using what's called the high-level Shading Language which Microsoft developed, a couple of years ago. What the High Level Shading Language does, it allows you to tell the video card how you want to vertices to be transformed. You can literally specify program code, to run for every single pixel on the screen, as they are being rendered. There is actually a program that can execute for every pixel so that you can make the lighting look exactly the way you want, make the -- you can do bump mapping, you can do normal mapping, you can do all kinds of very advanced graphics techniques in real time that the user clearly offline renders with modern video hardware, it's actually really easy to do a lot of this stuff. And HLSL, moves you out of originally the shading languages where assembly based languages and HLSL moves it up to a more C, C++ type syntax and then there is a Shader Compiler that runs and it generates front end code and then there is a back end compiler that generally the video card manufacturers tweak out so that you get better backend code and so forth as they tweak the drivers.

(00:40:05)

Carl Franklin: Wow, this is exciting.

Frank Savage: You had to have a card that understands at least the 1.1 version of that Shading Language, or our stuff doesn't even run. We immediately early out and say, "Listen, you don't have the hardware that can do our stuff." Those cards have been around since 2001 so chances are, even if you have a commodity laptop that you bought in the last two years, there's probably 3D hardware on it, you just don't know it, which is really cool. Most people don't realize that and that's -- this is helping to kind of drive home the fact that these parts are very ubiquitous now and you know what, you shouldn't be making casual games that are very sprite based and boring looking. You should be making more 3D oriented stuff and there is a lot of effects you can get away with today that have a very broad reach, but people don't do them, because they don't understand the hardware that is there and they don't want to go detect it.

If I'm a casual game developer with a three months development cycle, I don't want to spend a month and a half figuring out how to detect all the different video cards in Windows. With our stuff, you don't have to do that. We automatically do it for you and we can actually -- you can ask what shader models you have and we can just tell you, it's 1.1, and then you can go, "Great!." HLSL actually to the effect framework, which is part of what we wrote. We will actually go and you can have all the different shader model implementations if you will, in the same shader source file. So, you can go in and actually say, "Here is what I want you to do for a 1.1 card, here is what I want you to do for 2.0 card, here is what I want you to do for a 3.0 card." It can all be in the same source file, it will automatically -- our effect framework will automatically pick the right one based on the card you have, you don't have to ask anybody anything and you can just run.

Richard Campbell: So, what's in the Xbox?

Frank Savage: So, the Xbox has a Shader Model 3.0 part. It was manufactured by ATI. I know the ….

Richard Campbell: This is the 360 we are talking about, right?

Frank Savage: Yes, the Xbox 360 has that in it. We don't run on the Xbox 1.

Richard Campbell: So, in theory if I want to write a game then, if I'm working on these games, it is my best interest to have a 3.0 Shader Model video card.

Frank Savage: If you want to be able to run the Xbox 360 and see what it looks like on Windows, yes, but the cool part is that if you already have an Xbox 360, it's so easy to just get it running on the 360, that you can let the Xbox 360 be you Shader Model 3.0 card, and let your Windows machine be whatever it just happens to have in it.

Carl Franklin: I bet the difference is striking.

Frank Savage: It can be. The Xbox 360 GPU is really, really powerful, as we are coming to find out. So, we have example games, so none of this would be valid without having some kind of data to back it up. Since we shipped it in December, we have gotten a lot of games already done on it, which is very cool. In fact even as we were setting this up, I got another mail in my inbox pointing me to yet another video of a game that someone had written that on our stuff. Anyway, this particular game was a racing game, it was done by Extreme in Germany. It was one developer, two artists, six weeks, and we were able to get the game running at 1080p resolution with two X anti aliasing at over 30 frames per second in managed code.

Richard Campbell: Normally, Xbox games before XNA, this was all level C++ coding for performance. I mean every game developer I've ever talked to is so performance focused, that was always the issue. Now, you are saying, "I'm still getting 30 frames per second out of this thing on managed code."

Frank Savage: So, the tradeoff of having to get as close to the metal as possible for at least the game logic parts of the game are starting to fade away, because processors have become so powerful.

Richard Campbell: We just got so much horsepower to play with.

Frank Savage: Exactly, and on the graphic side of it, the shader stuff is -- again we shader compile it down to the actual microcode that the shader, that the Graphics Processing Unit wants to use, and the managed code side of it is really just setting a bunch of state in the Graphics Processing Unit. Essentially, what's going on is, we build up the command buffer that the graphics processing unit is going to use, we set all the state or the registers in that, and then we just say, "Execute," and then we are done. GPU then renders the stuff as fast as it possibly can. As fast as it possibly can, the Xbox is actually almost unbelievably fast.

Carl Franklin: Tell us about TorqueX. What's that? This is some Third Party stuff, right?

Frank Savage: Yup, TorqueX is a game engine that is being developed by our partners at GarageGames. GarageGames approached us last year at GDC, when we kind of unveiled some of the stuff we were doing, and was very enthusiastic about being an early adopter. They took all of their libraries and converted them to managed code, all C#. Got on the bandwagon very, very early. They now have a suite of tools and game engines and Data Form that you can download, that build on top of the frameworks and stuff that we already have. The cool part is we also announced at GDC this year that if you've bought the $100 subscription that allows you to program on the Xbox 360, and run on the Xbox 360 your games, you get a free license to the TorqueX engines and everything else as well.

(00:45:25)

Carl Franklin: So, they basically make it even easier. They add some sample code and some sprite stuff.

Frank Savage: Yeah, so they have a game building engine, if you will, that allows you to sit down and mockup a game very, very quickly, define behaviors, talk about all that kind of stuff. They have their own scripting language behind it, they have replaced that all with C#. It's actually very, very cool. So, this last GDC as well we had four people sit out in public -- in the lobby bar of one of the GDC buildings and those four people sat with their PCs and Xbox 360s making a game in four days in front of everybody. The press was there, everybody was there, and they were watching these guys make a game from scratch, from literally nothing using our stuff and the Torque stuff in four days. We got four relatively good games out of them.

Carl Franklin: When you say, relatively good, they wasn't like Hangman, was it?

Frank Savage: No, these were actually compelling and playable. It was actually pretty amazing. We took the games to the party that we

had that Microsoft has every year at GDC and had these four games setup and they were playing on this like 65 foot giant projection screen up on one wall and people, they were mobbed, the entire time. You couldn't move from the people who were non-Microsoft people who just happened to be at the party just couldn't believe that this much game had been written in four days using managed code and that it worked this well and how did you do this, and why did you do that. It was amazing, how fast you can get stuff done in this environment.

Carl Franklin: Tell us about the Creators Club.

Frank Savage: So, the Creators Club is the $100 subscription. We require a $100 subscription on the Xbox 360, that you pay as part of Live. So, your Xbox has to be connected to Live, you have to have a valid subscription and then our host will launch a run on the Xbox 360. If you've paid us that subscription, you are a premium member actually of the Creators Club. So, the Creators Club itself is just a community. We want to build up an enormous community of people who are actively using our stuff, give them samples, give them Starter Kits. I don't know how familiar your audience is with the concept of the Starter Kit and the .NET stuff.

Carl Franklin: Oh sure.

Frank Savage: But basically, we give you an entire game, all the content. You go into Visual Studio Express you say, "New project, Space War game" or a "New project, Racing game," and you get a whole Space War game and a Racing game. All the content is there, it's all lined up in Visual Studio correctly. We have content pipeline where the content is actually – as it fits as first class citizens with the code files in Visual Studio, so that it's very easy to get content into the game in terms of textures and models and meshes, shaders, all that kind of stuff. Audio, all that is treated as data files, it has to be part of the code as well and they've become resources that they are very easy to access to the framework.

Richard Campbell: I'm betting that Starter Kit was a whole bunch smaller than a gigabyte, like 'Mech Commander 2' was.

Frank Savage: Yes, it was. So, we shipped the GSE that's on the web today with -- The Game Studio Express download has Space Wars inside of it and that's probably the reason why the download is about 80 megabytes, is because Space Wars probably gets 70 of it. The Space War is a modern 3D implementation of the original Space War game, that was written on the PDP 1, and you can see 3D wise how the ships look way better. You get asteroids with real lighting on them, in the game you can blow up the asteroids, you can blow up the other ship, you can buy new weapons. So again, keeping with that concept of kind of a Geometry War as retro and evolve, we have a Space War Retro, which is just the old line based version of the game, and then the evolved version of it, where the ships are much better, all 3D rendered with all kinds of graphics and stuff. Again, all the source, all the content, everything you need to be able to dissect what we did, and really understand how this framework works in a real application if you will.

Carl Franklin: So, I'm looking at the XNA Developer Center which is on MSDN, Google XNA toolkit we got there. You have a link on the left that says, 'XNA Videos' and I noticed that you've got some tutorial videos here displaying a 3D model on the screen, making your model move using input and making sounds with XNA Game Studio Express and XACT. What is XACT?

(00:49:55)

Frank Savage: So, XACT is the sound system that we used for the framework that we have. We don't actually have any low level sound APIs there, so there isn't anything like just play this Wave file or play this MP3 file or something like that. Instead, the XACT engine uses a tool to build up all of the Wave files, WMA files, MP3 files, whatever you've got and it builds them into what are called sound in wave banks. Those are just big files that have all of the sound data put in them in a way that makes them easy to stream on the Xbox 360, so that they are not sitting on the memory all at once. Windows has an engine for that now as well, again for the same purpose, to keep you from having it all on memory all at once.

There is actually a separate tool, the XACT tool, which ships with our stuff as well, that gets installed, that actually allows you to edit those files and create new sounds and you can change reverb on them and do all kinds of great sound engineer kinds of stuff with them.

Carl Franklin: So, I have a question on how you're going to handle this, but I'll throw it out there anyway.

Frank Savage: Sure

Carl Franklin: What did you guys think about the popularity of the Nintendo Wii? Has that changed you guys at all, in terms of the way gaming is approached?

Frank Savage: So, it's made us in terms of the XNA community game platform stuff, significantly more exciting to a lot of upper management here, because it's a differentiator, it's something that Sony and Nintendo both will have a very hard time following up with. Not only do have the capability to program the console in a way that would be very hard for others to emulate sort of installing Linux and getting libraries and getting all that stuff to work and then realizing in a way, there's no actual contact with the graphics card or with the Cell Processors that are there. We actually give you all that on the Xbox again, minus networking today, but that's coming. But the other side of that equation is, despite the fact that we can now build all of these games, what we really want is this enormous community of game developers who are building them and then game players who are actually consuming them. We want to have this vibrant community of players who start to go off and actually download these games and play, instead of the tons of the games that are in arcade today, be able to play the thousands of games that the community can actually dream off and actually create. Significant fraction of them are probably not going to be so good, but there's going to be a lot of them that are really, really good.

Richard Campbell: Well, it doesn't take that many, does it?

Frank Savage: No, it really doesn't.

Richard Campbell: You've got this advantage of scale, there is thousands of people making games. The average human can only play so many games. Give me a couple of dozen that are great.

Frank Savage: Exactly.

Richard Campbell: You've got something going on.

Frank Savage: Yeah and it becomes a very different universe if you will to live in. For example, I don't have to wait for these great game ideas to get through a publisher and get done and everything else. So, the community can actually whip them out pretty quickly. We can see that if the great game idea -- at that point a whole bunch of different possibilities occur, the community goes out and builds it into a much larger, more compelling experience. A publisher picks up the game and pays somebody to actually go and finish it. We pick up the game and encourage it to move on to a Live Arcade or on to Silver Media and we'd publish it. There's all kinds of possibilities where today the game industry as a whole is -- it's hard to call it stagnant, because there's a lot of good stuff going on, but quite frankly it's a lot of the same stuff over and over again.

Carl Franklin: So, can you answer my question now?

Frank Savage: Sure.

Carl Franklin: I asked a question particularly about, what's your reaction to the Nintendo Wii? The controller which is completely different from anything anyone has ever seen before.

Frank Savage: Sure, but the controller itself isn't the differentiating feature, right?

Carl Franklin: It absolutely is.

Frank Savage: Is it?

Carl Franklin: Absolutely.

Frank Savage: What do you play besides Wii Sports with it?

Carl Franklin: Okay look, it's the most popular gaming console ever. People are reacting to this. I guess the answer…

Frank Savage: Well, people are reacting to it too, but I mean honestly, there are 43 million Playstation 2s installed, and Playstation 2 is still outselling Xbox 360 today.

Carl Franklin: I guess my question is, is there anything in the works to sort of go after the cool kind of controller stuff that they are doing?

Frank Savage: There are. Unfortunately, I'm not really at liberty to talk about that.

Carl Franklin: Okay, question answered.

Frank Savage: So, I can put it that way.

Carl Franklin: I guess that's what I was getting at.

Frank Savage: We certainly are interested in understanding why I think the Wii is as popular as it is. I think the controller is certainly a large part of it, but I think it's the controller married with an incredibly great software experience that actually is the differentiating feature there in exactly the same way that the Nintendo DS is a much more compelling game platform than a lot of other handhelds for exactly the reason that Nintendogs and Brain Age are very great meetings of that…

(00:55:06)

Carl Franklin: It's not about the realism of the graphics which is interesting.

Frank Savage: Exactly, it's about game play.

Richard Campbell: Nintendo has got a very interesting culture. They have a relatively few number of development teams it seems. They hold them much closer to the company, so there are much more company games. As opposed to Sony, who will sell just about anybody a development platform, and then there's very little control over the games. There's a lot of games, but a lot of them aren't that good.

Carl Franklin: Well, let's talk about controller development. Is that a possibility or is that left to the specialty boutique hardware shops to invent new controllers?

Frank Savage: No, I think we are going to see some controller innovation from Microsoft here in the not too distant future.

Richard Campbell: That fact that you are using USB as your connecting technology really opens the door to that.

Frank Savage: Right but I think -- well here's the difference though. So, you've got USB, that's attached to it, which means that yes, there's a significantly larger potentially amount of hardware that you can plug into it, like right off the shelf too, which is a cool part. All you really need is to write a simple little driver module, that catches the USB date packets coming off the port, and we just hand them off. So, that's a cool…

Carl Franklin: Right, there's a whole bunch of Japanese devices that you could probably plug in there, aye Richard?

Richard Campbell: Oh boy.

Frank Savage: Exactly. So, right out of the gate we have that advantage if you will. But the really cool part and the place where we come in is, again we don't necessarily want the game studios like your EAs or even Microsoft game studios to be worried about the new controller. Why don't we give it out to this community of hundreds of thousands of people, who can think of ways to use this that quite frankly we can't even imagine today.

Carl Franklin: I quite agree that the whole opening up of the development of these things to the community is what Microsoft has been all about in other development arenas. I mean that's what the whole .NET community, what makes it vibrant, what makes it work. This is just taking that model and extending it, and I kind of think that there's more parallels here. I kind of see the gaming software business as sort of accelerated software in terms of a way that we can look at our own future of business software. Like, regular software follows game software, because that's where all the innovation is, because that's where the hype is and all that. So, you can sort of see parallels there as well. It shows us what we are going to be doing in the future.

Frank Savage: Yeah, I think so. We are at the crust in game development now. It's kind of interesting, where we have a number of smart people who have been in the business for a long time or way outside of Microsoft, who pointed to us using .NET and C# and said, "This is the future." Game developers today are going to be resistant to this future, but they are going to be resistant in exactly the way that they were 15 years ago, when we told them that C++ was coming, and they were like, "No, C++ will never be performed enough, it will never be fast enough. I'm always going to write my game in C, you're never going to go to C++.

Richard Campbell: Or DirectX, they had the same reaction as well.

Frank Savage: Exactly, and today there isn't a game in the world that isn't written in C++.

Carl Franklin: Now, you can just write a few lines of C# and boom, 30 frames a second.

Frank Savage: Exactly. It's changing, and I think that's one of the ways that we're going to fundamentally change the business, we're going to give the average Joe the ability to compete and show up and show us their game idea. Show us what you got.

Richard Campbell: The challenge is that, modern games are developed with a 100 people like you said. Dozens and dozens of artists and another dozen developers. Can you really get that kind of quality out of that smaller group of people? Therein lies the challenge. Where does this art come from?

Frank Savage: Well that's the point though. It's not a really small group of people, it's a giant community. It's hundreds of thousands of people all contributing to this. What if my game were on some kind of public access place like CodePlex or some thing like that. And I was actively searching, hey look I need space ships. Go make me space ships.

Richard Campbell: Are we really talking about an Open Source type environment here, where everybody can contribute to anybody's game?

Frank Savage: Yeah, absolutely.

Richard Campbell: Doesn't sound very Microsoft-y, does it?

Frank Savage: No, it really doesn't.

Richard Campbell: It's not just community oriented in, "Hey, here's my game, if you like it go ahead and play it and maybe I'll make it shareware, you can pay me five bucks if you love it." But you're actually saying, "Here's my game and all the code."

Frank Savage: Exactly, and make the code better too. If you got a better idea for how the AI should work, please by all means, plug in.

Carl Franklin: So, with all these tools, you can obviously write more things than games for the Xbox 360. Like Media Center programs and things like that I can see. Do you see things moving that way for the Xbox?

(01:00:05)

Frank Savage: We do. Right now, the API's don't expose any of that unfortunately. That was one of the biggest requests we've gotten from the development community, is they'd love to be able to get to the Media Connect pieces of it, and be able to write their own streamers and players for their Media Center PC's.

Carl Franklin: But you said, the network stuff will be exposed pretty soon, you can imagine subscribing to RSS feeds, and getting BitTorrent clients on there and all sorts of great stuff.

Frank Savage: Yup, all of that is certainly possible depending on how open we make the networking, because we can also -- we can tie it so that the boxes can only talk to other boxes, that they aren't like general purpose sockets.

Richard Campbell: Give me an IP and I'll rule the world.

Frank Savage: It turns out that no matter how hard you try to secure that, as soon as you say that you can -- you have a debug channel backed to a Windows box, it kind of all flies out of the door anyway.

Carl Franklin: So, it sounds like you'd like to do that, but you are trying not to make upset any partners or business, like you said before, business models that work around that.

Frank Savage: Exactly, it's all about the business models and not cannibalizing a lot of the people who are actively making a lot of money, doing the Live stuff and stuff like that, we want to make sure that…

Richard Campbell: You've also made this platform possible. You can just mow them down along the way.

Carl Franklin: No exactly.

Frank Savage: Exactly.

Carl Franklin: But I can imagine that some of them have got to be a little bit put off by the opening up of this platform. I mean, you basically open it up to some serious competition, all the game developers, you know.

Frank Savage: So, the funny part of it is that the Game Development Community itself is very interested in using this stuff as well. So, to I guess kind of our surprise when we were at the game developers conference a couple of weeks ago, we heard from a lot of game developers, who'd been in the business for years and years, who had finished their Christmas titles, were looking for something to do, SAR Stuff out on the web, went and downloaded it, started playing with it and came to the end of January, getting ready to ramp up on their new titles. Instead of ramping up on a new title, they had a prototype done and working. And they were surprised -- was what we heard over and over and again, that how quickly and how easily and how performant they could get something up and running in an incredibly short period of time. We actually have two companies now that are actively building Xbox Live Arcade titles using this stuff, and they are getting ready to ship those out hopefully this summer, and they started as prototypes using the XNA Game Studio Express, got done with their prototypes and said, "Well, why would we move this off of C#, it's fast, it's performant, it works, it's the gameplay we want right here, why would we move it?

Richard Campbell: It's pretty exciting. It's very disruptive to the existing business models, smaller teams, shorter cycles. What are these big guys going to do to justify their existence?

Frank Savage: Well, I think you are still going to see in much the same way in the movie industry today, you've got a lot of direct a video, lower budget stuff going on. I think you are going to start to see that, see what the community does and the game studios are going to be able to concentrate on their strength anyway, which is the blockbusters. The biggest problem they have today, is they have to fund 15 start up blockbuster movies to get the one or two, that come out, that are actually worth anything, that they actually make all their money on. They won't

have the fund 15 startups ones anymore. They can find five, because they know that all the rest of the games are either going to come from the community, or they can go and leverage the community ones that are already done.

Carl Franklin: They got to think that, if four guys can make a game in four days, just think of what a big company can do with the resources that they have now with the technology they are using now.

Frank Savage: Exactly!

Carl Franklin: You can just kick them up a notch and you know. Can I have the full version of this game which comes on 44 DVD's? It comes on a hard drive.

Richard Campbell: We get into an interesting angle on this, which is that distribution costs around games are non-trivial. Let me show-off my geeky gaming shops and say, Where does something like Steam fit into this equation?

Frank Savage: Sure!

Carl Franklin: What is Steam by the way?

Frank Savage: So, Steam is the distribution mechanism that Valve created for its Half-Life product and actually a significant number of games, and at least 20 odd that I can think of using this as their distribution mechanism now.

Richard Campbell: So, you are getting these sort of Indy-ish games that probably wouldn't have made it on the shelves, but I can download them for 20 bucks via Steam.

Frank Savage: I think there is pretty clearly a business model around that, and what are the kind of things that helps drive content into that model, I think.

(01:05:00)

Richard Campbell: Well, especially if you can make it a fully public model, that I can develop this game, and test it and package it in this space and then say, "Lets roll it out to the marketplace, and sell it for five bucks."

Carl Franklin: Wow, disruptive is the word of the day here. So, how long has the Studio Express been available? You said it just started shipping?

Frank Savage: Yeah, we started in December of last year and we are getting ready to roll out the second release, which is more of an incremental release than the kind of giant one we just did. So, incrementally we are adding things like 3D audio support to the XACT Audio Engine that's already there. A lot of people have asked for, "Hey, you give me all of these 5.1 speakers, can I please just put the sound some place other than right and left?" So, we said, "Yeah that would be great." So, we added that.

Probably the biggest thing is the fact that right now, if I want to share my game with someone else, I have to share all of the source and the CSproj and everything else for them to be able to actually play it. They have to actually compile it, and in the process of compiling it and deploying it to their Xbox 360, it winds up on the Xbox 360 and then there is a browser on the Xbox 360 that lets me look through all of the games that I've uploaded to my Xbox 360, and play any one I want, but that initial deployment if you will, to my Xbox, that's still happening through Visual Studio. Of course, to do that, I needed to compile it and build all the code. In our April release, which is coming out here very soon, you will be able to actually bundle up all the pieces that the Xbox needs to be able to run and by double clicking on that bundle thing, it will actually deploy it to the Xbox for you.

So, you won't need the source anymore, you can actually create a package if you will, of a game that can be installed from any Windows based PC to any Xbox 360, as long as that Xbox 360 again has the subscription on it.

Richard Campbell: Right, so it's still a PC to PC distribution model?

Frank Savage: Yes, right now.

Richard Campbell: Someday, may be we'll have an Xbox to Xbox distribution model?

Frank Savage: Yeah, we are actively looking at what some of that looks like for the fall release, which is coming this fall, doing a lot more, a lot more around community and sharing and how does somebody without a subscription be able to play these games is the problem we want to solve for this fall.

Richard Campbell: Right, maybe a different kind of subscription that is just for getting access to playing the games rather than being a contributor.

Frank Savage: Exactly.

Carl Franklin: Now, here is something that -- if it's going to be so easy to create these games, I can imagine some high budget album projects that instead of bands releasing a CD or a series of MP3s through iTunes. They release an Xbox version of that, which plays the tunes through the

Xbox, lets you select them, and then has sort of these interactive art things going on. Like I think, Beck would have a field day with this.

Frank Savage: Sure, Absolutely.

Carl Franklin: Don't you?

Richard Campbell: I was thinking Oldfield, but yeah.

Frank Savage: It's one of the things that we have been talking about, I mean Origin built Wing Commander III as an interactive movie, and then the interactive movie thing kind of went by the wayside and everybody was like, "Yeah whatever, all games are that way now, and that it's not even worth building them that way, because they all have to tell some story, or no body plays them any more." I think we are at that cusp again, where the game experience is going to start to bleed over into other media types, where you …

Richard Campbell: What's a game?

Frank Savage: Yeah, what's a game, exactly and the pun here likes to argue that American Idol is a game. That it's not actually a TV show, or…

Richard Campbell: It isn't, Howard Stern is winning.

Frank Savage: Exactly, because people are actively playing, they call in and try to get the person they don't like off the show, and people game it, and yeah, by some definition that's a game, maybe they all are, maybe we are about to see that kind of -- this is one of the biggest thing is, how do we get, -- there is a fairly large community of hardcore gamers, and these are the guys that will stand in a line to buy Halo 3, and new consoles and those kinds of things. There is a fairly significantly larger audience, who play casual games, who play things like Hexagon Windows, or Solitaire or those kinds of things and those are all games too, and what's been difficult in the industry is, how do I monetize those kinds of very casual small experiences, and people like Oberon and Popcap, have figured out how to do that, and they make casual games now and they target at those audiences, and they make pretty good money doing that. I think that community is poised to open it up even further, because instead of having very targeted casual experience at a specific audience or segment, you are going to get these casual experiences that aren't really games that are going to blur that line even more and they are going to be the kind of things that people want to experience and play and download and do things with it.

(01:09:57)

Traditional business models are going to be hard pressed to understand how to monetize and we are spending a fair amount of time today in our own organization trying to figure out when that happens, not if, but when that happens. How do we monetize it? What are the kinds of things we can do to help make that happen, and make a lot of money from it, so.

Carl Franklin: Well, I think we have come just about to the end of our show here, and I want to close this show with a joke, because this has been kind of fun for me, and why not just go out with a bang. So, I got to tell you this new joke Frank. So, a couple of software guys, a couple of developers were talking in the cafeteria at lunch and one says, "Guess what, yesterday I met this awesomely gorgeous blonde in a bar." The other guys says, "Really? What did you do?" He says, "Well, I invited her over to my place, we had a couple of drinks, we got in the mood and then she suddenly asked me to take her clothes off," and the other guy says, "You are kidding me." He says, "Yeah, I took her mini skirt off and I lifted her and bent her over my desk next to my new laptop, and he says, "Really? You got a new laptop?"

Frank Savage: Yeah, I can identify with that.

Carl Franklin: Kind of brings it full circle from what we were talking about earlier in the show.

Frank Savage: My wife often brings up the fact that it's a wonder that we have children in our family.

Carl Franklin: Alright on that note, Frank Savage, thanks for talking to us today, and great job, congratulations.

Frank Savage: My pleasure, thanks!

Carl Franklin: And I'll be looking forward to messing around with this myself.

Frank Savage: cool!

Carl Franklin: And we will see you next time on .NET Rocks!

Carl Franklin: .NET Rocks! is recorded and produced by Pwop Productions - providing professional audio, audio mastering, video, post production and podcasting services, online at www.pwop.com. .NET Rocks! is a production of FranklinsNet – training developers to work smarter and offering custom onsite classes in Microsoft development technology with expert

developers, online at www.franklins.net. For more .NET Rocks! episodes and to subscribe to the podcast feeds, go to our website at www.dotnetrocks.com.