GENERAL

About ASCII World
News & Events
Recent Changes
Related Links
Bulletin Board
Staff/Contact
Page Tags

Helping Out
Want to join?

ASCII

ASCII code table
Standards
History
ASCII Art
Art History
Animations

EFFORTS

Asciratzi
Guess it
PCopy
Retro2FB
S.D.I.

ACADEMY

Tutorials & Lessons
Techniques

Interview With Richard D. Clark

ASCII-World » Asciratzi » Interview With Richard D. Clark

No matter where you've been in the QMunity no matter which forum you visit or why you visit them. One name always seems to appear there. A name that always seems to find the time to help you with your questions, to guide your programming projects in the right direction. That name always seems to be willing to set time aside to help in more than one way. He's created documents of all kinds concerning programming in more than one language. That name is rdc.

Just who is the man behind this three letter nick name? Find out exactly what makes him tick (so to speak) as we engage in an interview with him right here for you, the readers. So sit back and discover, as I did, who rdc is and what makes him such an awesome guy to talk to and work with.

E.K.V: *Ok, lets start with the "specs". Who, what and where?*

R.C: My name is Richard Clark. I live in Texas, USA (which is the only state in the union that used to be its own country btw: The Republic of Texas). I was a professional programmer for quite a number of years working as a consultant for companies like Mobil Oil, J.C. Penny, Form Motor Credit and ACS, along with a host of medium to small businesses. I primarily developed Windows application in Visual Basic using SQL Server. Fairly boring stuff, but it did have it moments.

With the IT downturn here in Texas, I have been working in non-computer related fields and have been concentrating on my writing. I write both fiction and non-fiction, with a lot of my nonfiction writing geared toward the programming community.

I started the book FBeginner: A Beginner's Guide to FreeBasic but had to pass it on to the community since I could not keep up with the changes to the compiler. Right now, I am working on a book about game design on the nonfiction side and working on a novel on the fiction side.

I also keep my programming skills up to date by working on small projects that don't take up too much time. Recently, I discovered the demo scene and have written a couple of demos and I am trying to learn some 2D graphics effects along the way.

E.K.V:*As an professional or retired programmer, how you see amateur scene of programming today?*

Concepts & Theories
Game Specifics
T.U.I. Specifics
Snippets
Miscalleneous

DOWNLOADS

Applications & Tools
Games
Demos
Libraries

ASCII-World EDITING

How to edit pages?
List all pages
Site Manager

Add a new page

new page

edit this panel

R.C: There doesn't seem to be much going on, at least in the programming circles I am a part of on a daily basis. It seems in the forums I frequent, there is a lot of talking about programming, but little product being produced. It is sad that the majority of posts in many programming forums are in the off-topic sections.

E.K.V:*You have also experience for different commercial and free programming softaware. What are those important differences between these two in use of normal "home programmer" and professional application developer?*

R.C: Generally as a professional programmer you don't get to choose your tools, you use whatever the shop uses. You may like mySQL but if the office uses SQL Server, you have to learn and program against SQL Server. This is both good and bad. The good side is that there is a consistent set of tools to use which make large projects easier to manage. The bad side is that you are stuck with those tools, even if something better is available.

For the hobbyist, professional tools are generally cost prohibitive. Who can afford $1000+ for a development package for just a hobby? I am sure there are people who can, but for most people the professional tools are out of reach.

However, there are a multitude of free or low cost tools available nowadays that the hobbyist can use that work just as good as most professional packages. Even Microsoft has released Express versions of their development tools that are free to download and use.

It is a strange situation. There are many more quality free tools available today then when I was programming for a living, and yet there is much less product available. I attribute this to the decline in professional programmers (at least here in the U.S.), which to large extent drive the hobby and open source market. I don't expect the situation here to improve anytime soon.

E.K.V: *Since you have programmed for so long now, it would be nice to hear when was the "golden era" of programming scene, in your opinion?*

R.C: For the hobbyist, the golden age was during the late 1970's and 1980's. There was an explosion of microcomputers on the market, the Vic 20, the C64 the Atari XL/XE/ST, the Amiga and Apple computers. There was also an explosion of software, everything from spreadsheets to games that ran on these small computers.

The games were of special interest. Since the computers had limited graphics capability, the programmers had to make the games interesting in order to be successful. Some of my favorite games appeared during this time like Lords of Conquest, M.U.L.E. and the Infocom series of text adventures. I bought just about every SSI game they made and all were of high quality.

When the IBM PC came out it was actually inferior to the microcomputer systems, but it paved the way for the computer to become a real office appliance. When the IBM compatibles appeared, the cost was low enough that the homer user could purchase one and it meant the death of the microcomputers, as well as the IBM PC. Most people nowadays that don't have an Apple computer are using what we once referred to as an IBM compatible computer. With Apple switching over to the Intel chips, they too are now part of the IBM compatible group.

There was a real sense of community back then. People would actually get together and meet face-to-face at SIG (Special Interest Group) meetings and information was freely exchanged. I was part of the Atari SIG in Dallas, and I really miss those get-togethers. The "communities" that exist today on the web today are pale imitations.
The BBS was king back then, and I remember how much fun it was to connect to the local BBS and chat with folks, download software and play the door games. People think the massive online games are something new, but there was a plethora of door games on the BBS systems that were ongoing and had hundreds of players. They were all ansi based, but that didn't detract from the immersiveness of the games. Most online, multi-player door games had much more content than the online games of today.

I remember connecting to the internet when it became available on Delphi, a dial-up service provider. It was text back then and we didn't have to worry about spammers, identity theft and viruses. Most of the information on the web was hosted at universities and all of it was free. And there was a wealth of information, both text and binary. For me, it was a sad day when corporations discovered the internet and had the idea to try and make a buck on it.

On the professional side, the golden age was during the 1990s, especially during the late '90s up to the year 2000. The web was taking off and companies were leaping on the .Com bandwagon in droves. The year 2000 was fast approaching and people were realizing that many, maybe most, of their software had the "Year 2000 Bug". There was more work than anyone could handle, and salaries were at an all-time high.

It is ironic that the web, which fueled the IT explosion, also brought about the beginning of its demise. When the .Com bubble burst and the stock market crashed, companies blamed the IT departments and the backlash was swift and devastating. I lived in Dallas at the time and I remember how many companies were folding up at an alarming pace. At one point there was a 60% vacancy rate in office space in Dallas; a sign of just how severe the crash was in the business world.

In my opinion, the country still has not recovered from the .Com crash, and never will. Outsourcing has become the method of operation for companies now, and although there are a few pockets of activity in the country, it isn't and never will again, be the the robust business it once was. The U.S. was once the leader in IT, but that is slowly shifting to Europe and Asia and that is where it will remain in the coming years.

E.K.V: *When you think about your programming career, is there anything you would change if you had the opportunity to? Anything you would have learned that you didn't or something?*

R.C: Not much I think, other than maybe doing more web programming and more graphics programming. Visual Basic, while not taken seriously by some programmers, was (and is) the best development system for creating business applications.

I managed to create some very sophisticated systems using VB, such as a rule-based expert system to score apartment applications and a distributed system that had components across the country all communicating with each other via a dedicated comm line. And everything in between.

VB was quite good to me and if I had to do it again, I'd probably stick with VB and application development.

E.K.V: *If you look at present day and maybe a bit of tomorrow, where you do see yourself as an programmer? Is programming only a hobby for you now on or do you have plans or even open doors to some professional projects?*

R.C: I don't expect to do any professional programming in the near future at least. Right now it is a hobby and I am enjoying working on my own projects, silly as they are. :)

E.K.V: *So, after years of programming career, coding can be fun with out intention to fill freeze with food? You havent had "ok, thats enough, no more computers" moments?*

R.C: No not really. It is my hobby, so I enjoy messing around with programs, games and of course I use it for writing. I'll probably keel over the keyboard one day and they'll have to bury me with it. :)

E.K.V: *When you are experienced programmer, is learning new things harder or easier than it is for beginners?*

R.C: I think it is much easier. No matter what programming language you use, the basic ideas are the same. The key to programming isn't understanding the syntax of a language, it is the ability to look at a problem, break it into its component parts and formulate a solution based on the different components. Once you understand the problem, then you should select the language that will make solving this problem as easy as possible.

A programmer should have more than one language in his or her toolkit. An arcade game may be easier to create using FreeBasic, but a GUI application is easier to create using Delphi or Visual Basic. A programming language is just a tool, and you should use the right tool to solve the problem at hand.

There is a tendency in programming circles for a language to become a religion. If you don't use XYZ language then you're anathema. The reality is that there isn't a single language that can do it all, and there never will be.

The beginner should start out, not with learning a particular language, but learning how to analyze the problem. Once they understand the problem, that is, understand the components that need to be built that when put together solve the problem, they can then shop around for the best language to help them solve that problem. You never start with the programming language; you always start with the problem. The language is just the means to the end.

E.K.V: *So newbie who's asking wich language would be best to learn, is kind of bit out of track allready? Maybe he should reform hes question?*

R.C: In my opinion, yes. How many times do we see a post on a forum about the next great game that someone is going to make (newbie or seasoned), and that is the last of it? The problem isn't the programming language, it is that the programmer doesn't know how to analyze the problem well enough to create a programming solution.

I would even say that this lack of analysis and clear objectives extend to the design and creation of the languages themselves in some cases. If the language designer doesn't have a clear plan on what they are trying to achieve, how can the programmer apply that language to a problem?

If I were to teach a class on programming, I wouldn't even introduce a programming language until the second semester. The first semester would be about how to think logically, how to analyze a problem, how to build and maintain an action plan and how to organize a project. Once you understand these basic concepts, using a programming language to build a solution will be much easier. This is the approach I am taking in my game design book.

Programming has always been 80% planning and 20% coding. The reason people spend so much time programming, and fail more times than they succeed, is that they do not have a clear plan when they build their solution. And often times when they do actually write the program, it is such a buggy, disorganized, unmaintainable mess, that you never see version 2 of the program.

E.K.V: *What do you think of the corporate world's evolution of software development tools? As in do you believe that the compiler makers (and other software development/analysis) makers have done a good job at keeping up with the new upcoming needs of the IT industry?*

R.C: The problem is that corporations, by and large, do not know what they need. You have desk monkeys that can barely check their email making IT decisions. The idea of proper analysis applies here as well. If you can't analyze your companies needs, you shouldn't be making the technology decisions, but that is the case in a lot of companies I worked for, especially the large corporations.

However, I think that by and large, the companies producing the tools have done a good job at extending the capabilities of the tools. As I said before, there is a wealth of programming tools available today that weren't available when I made my living as a programmer, and many of them are free or low cost.

In some cases the intent, such as the Express versions of Microsoft's programming tools, are intended to build brand awareness, but that doesn't detract from the fact that they are available for free with few or no strings attached. It is paradoxical that so many quality tools are available, and yet the industry is declining at an alarming rate. However, for the hobbyist programmer, these are good times indeed.

E.K.V: *We have those desk monkeys here as politicians too so i do know what youre talking about. Do you see any chance that amateur programming scene could really challenge professional side in future?*

R.C: I think it already has as evidenced by the Open Source movement. Even though many, maybe most FOSS software is created by professional programmers, most of it is outside the corporate channel and many companies such as Microsoft have felt the impact of the products.

The US lags behind Europe in OS adoption, but I think that corporations will look at OS much harder here as a way to cut costs which is always what companies want to do to boost the bottom line. They have cut the staffing to the bare bones and outsourced what they can, so the next step will be to adopt low-cost and free OS software to lessen the impact of the overbearing license schemes software companies have created here in the US.

One thing that is interesting in this respect is game software. The indie game producers are entering a renaissance I believe since many game developers have abandoned the PC market for consoles. This is a real opportunity for indie game developers on the PC if they can offer good game play at a reasonable cost. It is quite easy now to get into game development with some free and low-cost game engines available on the internet.

Even in the console arena, indie game developers have made a mark, such as in the XBox online marketplace. Many of the indie games are just as popular, maybe more popular than the commercial games.

The next big crash in the tech industry will be in the game market. What many people don't realize is that the game industry, especially the consoles, is cyclical. I have been around since the beginning of the console industry and have seen it happen several times.

The new breed of console hardware and software are so expensive to produce that the price points are reaching the limit on what the general public (where the companies really make their money) are willing to spend. Right now the shelf-life of an average game is about a week. Not nearly enough time to get a decent return. I expect the WII to do well, but I have doubts about the others.

One area of the PC market that will grow however is the online multi-player game. It is relatively cheap for the gamer to enter and offers extended game play. As long as the online game companies keep adding new content to their games, they will continue to do well.

Another area is the hand-held market, especially mobile phones. Since most phones are powered by Java or embedded Windows, and the games are easy and cheap to produce, this market will continue to expand. If I were to start a game company, it would be making games for the mobile market.

E.K.V: *What you would now say to yourself if you could give some advice to yourself in that age when you started learn programming?*

R.C: Just that as a programmer you need to expand your horizons. Nothing stays the same very long in the tech industry and the more diverse you the better it will be in the long run.

Now, we do know more about that name and thoughts behind of it. I wana thank rdc about this interview.

E.K.Virtanen

e.k.virtanen interview rdc