

[Please click here for the latest colorForth information.](#)

colorForth and the Art of the Impossible

Howard Oakford, Inventio Software Ltd, 19 Oct 2001
www.inventio.co.uk

"I choose a block of marble and chop off whatever I don't need" Auguste Rodin (when asked how he managed to make his remarkable statues).

Abstract

The direction of computer science in the early twenty-first century is the same as the culture which is creating it – global, standardised uniformity. How refreshing to find a new programming environment which bucks the trend.

ColorForth

Chuck Moore described his ColorForth as "brutally simple". Using it has been compared to "programming an alien computer". It is gloriously non-standard.

- ColorForth uses only 27 of the 102 or more keys available, and the letters are not the ones printed on the keyboard. The Enter key is not used, and the space bar is used to enter a command.
- The font provided as standard is the conventional Latin alphabet, but is not in alphabetical order, and can be modified by the Icon editor to display any pattern you like.
- Characters are stored using Huffman encoding, not ASCII. Yes, ASCII is not used. 'a' is 5, not 97, 'b' is 19, not 98. Definitely not ASCII...
- The text editor stores pre-parsed Huffman encoded strings (one or more 32 bit values) into Blocks, and must Huffman decode before displaying them.
- One call to the BIOS to set up the video card is all that remains of the layers of operating system which usually cripple your PC.

So you've found a computer with the right sort of AGP (or PCI) video card, and (in my case) dug up an old multi-function ISA card with a compatible floppy disk chip and got colorForth to boot. What next?

The bottom right hand corner of the screen displays a table of the functions of the 27 active keys on your keyboard. Press "e" and the enter key, (or "d" and the space bar as they are more commonly known,) to start the editor.

Something like a Forth block appears on the screen, but without the line numbers, and in lots of different colours, and the keyboard pattern changes to show you the new set of functions which define the editor. "-" and "+" (the "M" and "/" keys) move up and down the blocks 2 at a time (documentation is on odd numbered blocks).

Having viewed the source, exit the editor (by pressing the space bar), and load the Icon editor by typing "icon" . You can now edit the font.

If you edit the pattern for a character which is currently being displayed on the screen you will see all occurrences of that character change. This is because there is a display task which continuously executes the Forth code following the word "show".

Everything you see on the screen is being continuously updated, at 24 Hz in my case, synchronised to the video display frame rate.

This is your opportunity to edit all of the character set to say, the Greek or Cyrillic alphabet, if that is what you are more comfortable with.

Type "save" to save the system to disk, but keep plenty of backups!

So you've learnt how to type using only 27 keys, to edit in colour, to setup a new keypad and actions. What next?

I visualise this new Forth environment as an infinitely large granite table, thinly marked out with a square grid. Each square is a Forth block. Some of the squares are filled by little islands of coloured text - these are the programs. One square is "home", where you start from. When you first run colorForth, this home square is the Forth outer interpreter, something like QUIT. If you type a Forth word's name here, you will execute it.

When you run one of these programs, you "jump" to that square, and you then see the world from there. Lets jump to the editor. The keypad actions change, and a new keypad display appears on the screen. The outer interpreter is no more, just editor functions. Now jump to the icon editor - yet another keypad design and set of functions.

ColorForth has several innovations - circular stacks, the splitting of the @ and ! atoms, the use of ones' complement subtraction and other improvements which have come out of Chuck's hardware designs.

But for me, the big one is the way that Chuck has applied the Forth "philosophy" to the Forth language, and created an environment which connects program context to the human brain in a simpler way :

The keypad display is a visual mnemonic for the program context and keypad actions. These actions have all of the possibilities that we are used to in Forth, but live in a world with even fewer boundaries.

The Art...

In conventional computer languages the system is divided up into functionally different components – editor, compiler, linker, debugger, emulator etc. An interface is defined between each component, and often this is a file containing ASCII characters. ColorForth has utilities to create such a file, in order to interface to the rest of the world, but does not use this format internally. It is easy to overlook the magnitude of this last sentence.

What are the advantages of discarding so much that is taken as fundamental by the rest of the computing world? This is in fact the wrong question, because it assumes that the decision to use or discard each and every standard was made on the basis of its relative advantages or disadvantages, taken separately. If you start with that mindset you will discard colorForth at the first hurdle. It is discussing the pile of rubble on the floor, not the work of art.

ColorForth is the way it is because it is necessary to be that way. It has its roots in the reality of its own environment, not in layers of artificially constructed interfaces.

...of the Impossible

Suppose for a moment that a computer removed from a crashed alien craft was powered up and its software analysed. What would it be like?

The keyboard would be more likely to have 27 keys than 102. The QWERTY keyboard has evolved from mechanical typewriters which needed one key for each character. If the aliens are carbon-based life forms they probably have relatively few digits, like us, so 27 keys would be more than enough.

The use of two stacks and multiple small procedures (i.e. Forth) can be discovered by any sufficiently advanced civilisation as a good way to write programs, likewise Huffman coding is a universal expression of data compression using a binary tree.

ASCII is just an Earth-bound agreement. The “operating system” approach with its manifold API's is merely an artefact of 21st century Western business practices.

Is it possible that two different civilisations, separated by time and space, could discover the same reality? Could ET have created colorForth?

Conclusion

Many people have been using Forth to create programs which are impossible with any other language. ColorForth is one of those programs. I like it.