READING ASSIGNMENT 02

In C. Reas' article I understood the common 'anatomy' of code and how its evolution together with the concept of 'software' has influenced the technological and everyday world in general. Actually, I have understood that there are coding languages with common characteristics: they are based in a specific algorithmic plan, which contains variable or constant quantities and commands which may use them either once or iteratively. They could be parallelized with Romance languages which all share similar basic structure and grammar, they are derived from the Latin language, but are different. Code usually does not resemble to a human's but to a robot's and needs enough accuracy for the computer to understand it – as a result, one could say understandable from almost every human. That was not the case though when code was born – the code was totally 'encrypted' for unskilled people and it was 'ugly', unintelligible computer jargon; it was firstly used for military purposes so aesthetics were not necessary at all. That changed as many 20th century artists shifted their interest revolutionarily to aesthetics by producing their works indirectly: by making a series of orders enough for an unskilled person in order to execute them. Also, there were efforts to make coding more friendly and personalised for a broader range of professionals and amateurs, while also make using computers suitable even for children, with the creation of games. A more important objective though was to enhance the appearance of the output on the screen and multiply the ways to modify it and the speed at which it is done, while also find new means to realise this – the Sutherland's 'Sketchpad' was one of the first and crucial milestones here but still not substituting drawing by hand. The next revolution though has achieved that with the advent of new devices utilizing laser technology, like laser printers which could transfer drawings from computer to paper: print them. Later with smarter technology geometry would become modifiable through the computer because mathematical principles and thinking would be integrated to it; nevertheless, there was still the concern for the quality of depiction. A solution to this was found with the creation of raster images and the decreasing of the size of the pixels of which it was composed; as a result, the screen seems today so 'accurate','sharp' and realistic -if needed- that it fools the eye. What about zooming-in or zooming-out our screen though? Because then the resolution would be 'spoiled', screens would better be created with vector lines where every bit of the screen would be variable and changing according to equations. Then the interest shifted to colour and realistic depiction: on the one hand, colour could be added in form of parametrical values to the value determining each pixel's colour; on the other hand, emphasis would be given in imitating how nature influences materials (e.g. make them seem dry, rough etc., apply photoshading etc.) and simulating light via oscilloscopes, a variety of textures but also how we perceive from our point of view an object (camera mode was added then). As well printers have got more accurate and flexible and now there is the choice of laser, inkjet or even 3d printers creating 3d shapes through various methods.

Today I believe that there has been an effort to adjust digital screens everywhere, hence coding and hardware engineering: from mobile phones and tablets to VR masks and digital glasses. There is also research to find ways to go outside the boundaries of the screen, i.e. to make our senses no longer limited to it. Such an example is a maybe hypothetical revolution in the sense of touch: holograms, which might seem just science-fiction; however, intense researches are in progress for creating them: one of them examines a machine which emits bounded light directly to the air without a need for a screen. The shapes composed of light could be dragged, scaled, moved or clicked by touching at strips of silicon. Because this would be a revolution in space depiction, it would surely help architectural representations.

As for artistic representations in general, there are sufficient programming languages suitable for such usage, like p5 and Processing. There you could create your own drawing tools in contradiction to CAD softwares where tools are ready-to-use. What if we could use them in holograms? That would possibly give opportunity to add new maybe stunning features to the existing languages.

With the advent of more advanced computers someone from 2021 might say: could computer technology grow more than that? I would argue ‘yes’! We might know about putting digital screens on mobiles, on glasses, watches, masks (VR) but the idea is not constrained on putting screens everywhere. Take this as an example: quantum computers which could be our century’s breakthrough, though not replace regular computers; there is also AI which, for me, could, sometime, stand as an ethical problem. If we look at Sutherland’s machine and then account for today’s technology then we might fear for the future: the former seems primitive towards AI’s like chatbots or recommendation systems in apps, so, if we turn ourselves to the medium-term future, then AI might be considered ‘primitive’; so now, what could be next? Could professionals like programmers be replaced by AI? Although AI could generate code, I would agree with Bill Gates as for his following notion: "The most amazing thing about software is that it allows people to do things they couldn’t do before, and that’s something you can only do if you’re a programmer.". And the programmers’ key role relies on what architects could rely on: creativity, their creativity, not AI’s creativity because perhaps there is no such thing.

Actually, a thing that impressed me was the road to the privatisation of programming: once it was just for military usage in the Second World War but now any people could work in the context of programming. Also, I remember the time I first played the videogame Minecraft when I was a kid; my first response to playing it was wondering: how is this game created? Then of course I could just guess how this game is created; in some way, the nature of coding softwares in general becomes somewhat inherent in new generations. Studies suggest that younger generations could evolve themselves better than older ones in programming software – hence in using it. One could say it is the nature of the slang of teens which might justify this ease because it somewhat resembles to a programming language. Anyway, how do teenagers in our days scroll, tap and type as if they were juggling, sometimes without seeing their device at all? I might say that usage has perhaps become a mechanical process done subconsciously.

As a summary, computing and coding has indirectly influenced our life from many aspects and in a relatively short period of time: Given the facts of earlier times, perhaps such rapid technological development would have been rare or even impossible. Nevertheless, I believe that programming could serve the needs of architecture and arts related to it (mainly spatial perhaps). I would say that the former is the 'best friend' for architecture and the prospects for both sciences in the future are remarkable, unless technology continues evolving at the pace it is now developing or even faster.