

The Importance of Multifaceted Skills Building

Community STEAM Education Models

[Computer History Museum](#), Ljubljana, Slovenia

Gaja Zornada, gaja@racunalniski-muzej.si

I will use an example from our community education work at the Computer History Museum to demonstrate how to use creative practices to engage and drive interest for science, technology and engineering even when the audience you are reaching out to defines itself as “not good in math”.

But first I want to explain why we feel it’s important to address young people who do not identify with STEM and not just those who already find science and technology interesting.

And why we believe no skill today can have only one side to it.

“If you put your mind to it, you can accomplish anything.”

– Marty McFly, Back to the Future, 1985

I’m a... **scientist**...turned **artist**... turned **manager**... turned **technology entrepreneur**... turned **innovator**... turned **educator and mentor**.

Our world is run by labels. And the most frequent label is that of a person’s job or profession. If you ask a child what they’re going to be when they grow up, you get a myriad of answers: “a scientist and a ballet dancer and a pilot, but also an explorer and a comics books writer”. Then they grow up and they end up being one thing. Isn’t that a bit strange? I guess that in that perspective I stayed the little kid.

Even after my successful career in the IT sector and a tech innovation track record, or as the head of the Computer History Museum, I find it often really puzzles people when they hear my educational background is as an artist.

But I know from my personal experience, there is a world of riches to be gained by really interlacing the A into STEAM. And so do many of my colleagues who studied film, philosophy, architecture, languages...and became computer programmers, product designers, QA engineers. Most of all the A has the power to make you feel at ease in ever changing context.

Prepare the young for life in **ever-changing** context in a world where the advance of science and technology already surpassed the limits of intuitive understanding.

We spent decades fencing off STEM education from the other subjects, the soft ones, dealing with human behavior, interaction and effects. And vice-versa.

And inadvertently we have done two things:

1. Turned too many people away from science and technology
2. Made the field of science and technology in public perception at least cryptic if not entirely inaccessible



We've produced many specialists who are experts in their narrow fields but lack the agility of point of view switching and broader implications interpretation. Skills which are already in high demand today, but will be even more sought after in the future, when our work and life will be dramatically altered by the very science and technology we do not understand intuitively.

Dall-e self-taught GPT3 AI produces this output when prompted to create an image of:
"An astronaut riding a horse in space as a pencil drawing".

It's not just about getting more girls to code. We need to get more of everyone. But how do we even get them through the door?

Software programming is one of these fields where deficiencies of narrow specialization and a disconnect from humanities and social sciences have become very apparent both in the talent pool and in products with painful consequences in coding bias, discrimination, addiction and so on. We are reminded on a daily basis how important it is to raise awareness and mitigate problematic behavior in young people, but most of all to raise them into a society of active co-creators of technology.

That is why we need to diversify the pool of people entering STEM fields. By doing that, diversifying the pool of personal experience, secondary skill sets, approaches to problem solving, and ability to shed light on various broader implications of technology on human life.

You've been attacked by a dragon!!! What is your next move?

3 hour workshop:

+ creative writing / storytelling

+ improv theater

+ coding

End result:

A live text adventure computer game based on the pure power of group imagination.

This is a pilot program we first performed in 2019 to reach out to profiles of students who would otherwise not be attracted to the offering of a Computer History Museum and spark an interest in technology for them through creative arts practices.

It was designed as a test program to see if we can get a non-STEM first audience into our programs if we make the A in STEAM really stand out. So we designed and communicated the program with a pronounced creative angle (creative writing and storytelling first, improvisational theater second and only then coding).

An interdisciplinary class automatically formed using different incoming motivations: storytelling, improv, coding.

We managed to attract a totally different set of students than with other coding programs. We realized that there was a present curiosity about coding in all of them, but they didn't feel comfortable joining groups who were too rigidly defined only through engineering. Which made us change how we approach all of our education programs.

Feed the dragon some **raspberrypi** ice cream?

```

3 </head>
4 <body>
5
6 <!-- story -->
7 <!-- story data -->
8
9 <!-- story title -->
10
11
12
13 <!-- story content -->
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86

```

Teamwork / Team building / Connection / Communication

Individual and group expression / Peer 2 Peer learning

Creative ideation / Brainstorming / Presentation

Selection / Evaluation / Compromise

Dramaturgy / Sentiment / Language registry / Rhythm

Analytical reasoning (if / then) / Program architecture design

Code structure implementation / Coding commands / Use of programming tools

Quality assurance / User testing / Bug fixing

The purpose of this literary - improv theater - computing workshop was to use interactive simple tools to create digital storytelling and show them that making a computer game can be a piece of cake.

And that all you really need is a bucket full of imagination and a systematic approach to your work. At the same time we wanted to spark interest in critical observation of the digital medium and empower them to tell a story in a different way and use technology for what it is - a tool.

The next generation of coders will need to think about societal values, because AI will not.

We need to get more of everyone to code to balance things out and build a techno-realistic future. Because the next generation of programmers will have a groundbreaking role for the future of humanity.

The programmers input will be redirected into code prompt engineering - creating the basic parameters and narrative for artificial intelligence to actually code the software. This means a key skill will suddenly become the correct formulation of the initial sentence of code for AI to finish it properly. Software creation will become faster, but understanding of the complete software stack will become even more elusive.

AI learns from large samples of program code and much of AI's behavior will be determined by the specific intricacies of the sample it uses for training.

Mistakes and assumptions will be transferred from program to program without deliberate human intervention or our knowledge and ability to identify it during the creative process. Software quality assurance, ethics monitoring and conscious shaping of social effects will become important parts of software development.

“It’s a miracle that curiosity survives formal education.”

- Albert Einstein

It really shouldn't be a miracle for curiosity to survive formal education. But if it is, we take it upon ourselves to fight this trend and make young people curious about technology, nature and society, because they will only have a future if they learn to merge all of those into a coherent effort to build a more advanced society.

And there is an added benefit if we teach them to be cognitively ambidextrous in the process, to innovate and create but also at the same time to structure their works systematically in an organized output. Those are the multifaceted skills they will need in the long run.