Code Awareness: Rationality

Code Awareness was born out of a desire to bring excitement to the field of programming, to make team-work feel less like task splitting, and more like eSports fighting.

## Social impact.

One of things I’m still working on, is trying to determine the effect this system would have if applied at large scale, throughout the world. The questions I’ve been trying to answer, or rather trying to gather enough knowledge to not spit out random statements:

1. Can this bring more people into the programing field?
2. Can this reduce the number of jobs available in the programming field?

One thing that I envision is a completely different training scene.

At first you start with a very simple stage, just assembly language (yes, assembly) and you learn about moving bytes around, getting smart about overcoming memory limits, switching bits on and off to trigger a series of light shows, timing interrupts perfectly to generate a little song.

 Then you move on to your first robot fight. Your robot has limited memory, but you can upload routines in real time, to make the robot adapt to the environment. This can go from simple routines meant to block attacks, to more complex routines that already have decision making factors embedded, up to the most refined routines that only office guiding principles to an AI powered robot. This scale can expand quite a bit.

Or perhaps it’s not a robot fight, but a harmony contest involving sensing sounds and vibrations. Or a light show, where participants have to adjust in real-time the position, color and timing of each laser and mirror, to match the surrounding teams and thus expand their territories. The world is your scallop... Or something like that.

That’s for training, for prize winning, for audience enjoyment, and of course for money, just like any sport.

 However, this sport is directly related to the performance of businesses, who will require more and more skilled programmers, adept in more fields than just logic. The businesses should actually sponsor these events, and thus capture the winning teams.

As for question #2, it is true that a more efficient system will produce better products, faster, will fewer people. However, this has always been the case throughout history. We’ll simply have to find better ways to create meaningful jobs, better ways to acknowledge diversity in effort and timing. There are other factors too which tend to reduce our job creation capacity, and we’re already experiencing such factors, hence all the redundant and convoluted office work. For programming at least, there is a lot more space to grow than in other fields. We only have to be careful where we build our fences.

The games can be quite creative, involving not just programmers, but hardware technicians, light and sound, dancers. Ahhh, the dancers are such a spectacle in my mind. The throw bubbles and arrows of light at the spectators, in the rhythm of the song they create on the spot, together, inducing an ecstasy never before experienced.

Today i was talking to a new friend about this game concept. He pointed out that my audience is a niche market. I agree, but it’s not just one niche market, there are several: music, painting, structural designers, architects, roboticists, aerospace engineers, pretty much any profession that has computers as one of its tools. I’m saying, you don’t have to restrict the input to raw programing code.