Your on the Street Reporter

Uyless Black

The Day Rocky Marciano Took a Dive

The Day Rocky Marciano Took a Dive

March 18, 1983 (with updates in 2005 and 2016)

Like most youngsters, my son Tommy likes to play games on the computer. As he became familiar with a game, he developed a sixth sense about how to execute the computer keys associated with the playing of the game, and consistently cleaned-house on his old man.

Yet during these early years, Tommy had no idea about how the games operated, or for that matter, how a computer worked. His mother was also a computer neophyte and held the view that he was destined for computer greatness because of his acumen with computer-based games. I explained to his mom that Tommy's resume would not stack-up very well against his contemporaries who understood the software that manipulated the games and the hardware that supported the software. Tommy was adroit with a mouse and his reaction time on the keyboard was impressive, but other than these superficial skills, my son was a computer illiterate.

Before long, Tommy also recognized his expertise in *playing* Dungeons and Dragons on a computer was not the same as *programming* Dungeons and Dragons on a computer. Thus, in his early teen-age years, Tommy began to learn about the details of software.

Earlier in my life, after almost committing myself to law school, I became fascinated by software programming and decided to make it my profession. I marveled then, and I marvel today, at the elegant logical order and composition of well-designed and well-written software. On occasions, I still read the programs I wrote some thirty years ago, sometimes admiring a clever subroutine I wrote, or admonishing myself when I come upon programming statements that were poorly coded.

Anyway, I was hopeful Tommy's interest in computers might lead him into the programming profession, or at least to an understanding and appreciation of computer automation.

Getting Started

When Tommy was fourteen years old, he studied the BASIC programming language. In conjunction with his new hobby, he and I spent several hours discussing the logic of coding BASIC for a mailbox application. Tommy had begun to use email and was fascinated by its ease of use and speed when he sent and received messages with his friends. During these discussions, I learned my son possessed a keen analytical mind, one that could translate ideas about an application into software code. As I played the role of his mailbox customer, Tommy queried me about my needs for the application. Changing roles, I would take on the part of a programming team leader, and Tommy would present his ideas about how to satisfy the customer's requirements by writing BASIC programs for the service.

These projects were not quite sufficient to cement Tommy's interest in programming and software. He liked the challenge of working with software, but he was still fixed at the user level. He remained a game-playing demon, but with only tangential interests in the innards (the coding) of a game. Later, he attended some college classes where he did very well in programming.

Heavenly Valley Ski Resort

I think the bait that made Tommy take my programming hook was placed before him during one of our skiing vacations. I had just completed a lecture about packet switching in San Francisco. After the lecture, I motored down to San Jose to pick up Tommy, who was living with

his mom. We then drove to the Heavenly Valley resort for a week of skiing. During our trip to the resort, we continued our discussion about the email application and took this opportunity to delve into the details of mail server functions.

During this drive, I was pleased to learn Tommy possessed a systems analyst's intuition. He was aware of the importance of clearing-up ambiguities about a customer's requirements for an automated system, and he understood how those requirements were translated into the functions and operations of a software-driven application. I learned that Tommy understood a customer's vague statements about his or her requirements translated into vague (sometimes incorrect) code.

Enter Rocky Marciano and Sugar Ray Robinson

Tommy had brought along a computer boxing game to play during our after-ski hours. Compared to today's fantastic offerings, the 1983 computer games were very primitive. This game displayed on the computer screen two fighters in a boxing ring---almost stick figures---who fought each other based on profiles Tommy and I entered into the program. Certain profiles worked well against other profiles, and some did not. For example, a profile of a fighter who liked to use his left hook made this fighter vulnerable to the opponent who possessed a strong, fast right jab. With fifteen variables available, Tommy and I were able to build relatively diverse profiles from one bout to another.

The game was programmed for one contestant to use keys 1-4 on the computer keyboard for manipulating his boxer's movements and fisticuffs. The other contestant used keys 7-0 for his boxer. Tommy warned me he had been playing the game for several weeks and was the champ of his neighborhood.

After skiing and dinner during our first two days at the resort, Tom and I played a few matches. For each fight, we would enter a different profile for our fighters, and then duke-it-out for ten rounds, or conclude our match when one of our fighters scored a KO.

We were also allowed to choose names for our fighters. I called my boxer Sugar Ray Robinson. Tom named his fighter Rocky Marciano (for the picky reader of this story, it made no difference if our fighters were in different weight classes---the names were non-operational variables in the BASIC program).

No matter what variables and names I used, Tommy routinely put my fighter down on the mat. I was determined to give Tommy and Rocky a match, but at this point, I did not know how I was going to compete with a teenager's prowess with a computer game. As computers and software evolved, so did our children's neocortexes. Kids' brains became programmed, like a software program, to trounce their parents and other adults when contesting on a computer game.

Revenge of the Parent

One afternoon I returned to our hotel room early while Tommy continued skiing on hills beyond my skill level---the black slopes. I turned on the computer, opened the fight game, played with it for a while, and discovered it was written in BASIC code. As I lamented earlier, I had been losing the game's fights with my son---I had not won one solitary victory---so I decided to even the playing field in my favor.

The figure to the left shows me laboring at the PC during this trip. The photo was snapped by Tommy, who did not realize he was about to be part of a carefully contrived joke. Notice the size of the "small" personal, portable computer. It was the revolutionary Compaq portable PC that transformed the computer industry. Also, notice my dark hair and, at that time, relatively small waistline.

During Tom's absence, I made a few changes to the BASIC code:

- The program continued to accept our input for our fighters' profiles.
- However, I coded Tommy's profile to be non-operational. The values he input from the keyboard were not used in his fighter's profile variables. They were displayed when he asked to see Rocky's profile on the computer screen, but they had no bearing on Rocky's performance. Unless Tommy read the BASIC code (which I knew he could not do, because I locked the code with a secret password, and he never thought to do it in the first place), he would not know he was the victim of my prank.
- To compound Rocky's humiliation and my son's frustrations, I programmed Rocky's profile in hard code, always vulnerable to any profile I might enter for Sugar Ray. In other words, I programmed Rocky Marciano to be a wimp.[1]
- The result of my hacking was that every time Sugar Ray threw a punch, he scored a knockdown on Rocky. Every time Rocky threw a punch, Sugar Ray countered, and put Rocky down on the mat. After a few knockdowns, the next punch from Sugar Ray would result in Rocky being KO'd. Rocky had become a hapless boxer. Sugar Ray had become…well, Sugar Ray.
- Sugar Ray and dad were ready for Rocky and Tommy.

Rocky Takes a Dive

I could not contain myself to prolong the joke for our post-dinner entertainment. After we cleaned up, I suggested we engage in a boxing game before eating. Tommy was agreeable to my challenge. After all, Rocky had pummeled Sugar Ray the past two evenings, and my son was out for more blood. He was convinced he had set up his profile to make Rocky a fighter who could not be defeated---especially with the adroit manner in which he manipulated the computer keys to move his fighter around the ring.

So far, Tommy had succeeded, but a change in fortunes was in the air. We sat down in front of the computer and opened the boxing game. I made inoperable changes to Sugar Ray's profile, while Tommy entered no adjustments to Rocky's capabilities. After all, he thought he had Rocky tuned to be an impregnable fighter.

The first fight was a debacle for Rocky. Nothing Tommy did with the four computer keys had any bearing whatsoever on the match. Rocky was KO'd before the end of the first round.

Tommy was somewhat surprised by Sugar Ray's victory, but he assumed I had made adjustments to counter Rocky's profile. We agreed on another fight. After all, our fighters had

[1] Forgive me, Mr. Marciano. You were my boxing hero when I was younger. But my son chose your name for his fighter, so you had to take a dive.

not yet broken a sweat. Tom changed Rocky's profile, but I announced Sugar Ray was staying the same. Why tamper with success?

The second fight was another rout. So was the third fight. So was the fourth. I was doing my best to suppress my knowledge of my hack, but I also did my best to tease my opponent about his pathetic performance.

I had my son befuddled. It was one thing to lose a match, but four KOs, in succession, all in the first round? What was worse, Rocky could not lay a glove on Sugar Ray. In fact, as the fourth bout came to an ignominious end for Tommy and Rocky, I realized I had likely over-programmed my kill to be over-kill. Yet Tommy had not yet become skeptical of the fights. He was (and is) not a suspicious-type person, and after all, his opponent was his own father (I suppose I am not a perfect role model. Scams on descendants are fair game).

Tommy could not figure-out how his exalted mastery of the computer boxing game had come to such an abrupt and ignominious end. The mystery was the subject of our dinner conversation, and I enjoyed every sentence of the discussion. Tommy was perplexed but eager to return to our room in order to experiment with Rocky's profiles and setup his fighter for a comeback. Fat chance.

The bouts after dinner followed the same scenario as the pre-dinner fights: a luckless Rocky Marciano succumbing to a fantastic Sugar Ray Robinson. I might add that my son was very impressed with his father's quick assimilation of computer games, especially in view that I was an adult---a category that should have disposed me to frequent defeats.

The Scam is Exposed

However, when we were into the sixth or seventh fight of the night, Tommy wised-up. He finally looked to me and said, "OK, something is wrong. You are much too smug about it all. I know! You have done something with the program---you must have changed code this afternoon. Come on, fess up!"

The ruse had worked, and I was ready to expose it, "Okay. I altered the code to give Rocky the boxing skills of Liberace. Want to see how I did it?"

"Absolutely."

I unlocked the code and displayed the program on the computer screen, "We are given fifteen variables to tailor the profiles of Sugar Ray and Rocky. A ten for each variable is a value for the best profile and a one represents the weakest."

Tom was eager to move-on, "I know. And we cannot build a profile with a total value of more than 75 points, so we must compromise and key-in values with care."

I continued, "Right. Now, the program is pretty sophisticated, and I was not able to dissect all its logic in an hour or so. But the person who wrote this code was a fine programmer. He or she wrote most of the code to be stable and simply coded some algorithms for applying the fifteen variables in a variety of combinations. Therefore, all I did was substitute fifteen pre-set constants in place of the fifteen variables for your fighter. Let me show you my changes."

I showed Tommy where I had made fifteen changes to the BASIC variables for his fighter. Again, the code was well-written. The programmer kept the logic simple and clean. He or she did not create any unnecessary temporary variables. Consequently, my changes were easy to debug, and I knew my substitutions worked after I ran a few tests while Tommy made his last runs on the ski slopes. I showed Tommy a few more aspects of the code, explaining how the values he saw on the screen when he asked for a profile were different from those used in the formula.

His observation was, "Dad, the program structure is really simple, yet the formula looks pretty complex."

"Yes, it looks like the programmer is using some probability logic, weighted by the values we input into our fighters' profiles. Here…you can see the values I hard-coded for Rocky."

Tommy looked at specific parts of the code, "Thanks a lot, dad. You hard-coded every constant for Rocky's profile to be one! …Except one constant. It is entered into the program with the maximum value of ten. What gives?"

"That constant is used to determine Rocky's ability to take a right hook. I just could not code that value to be a one. Rocky Marciano had a steel jaw, especially for hooks."

Whatever my silly rationalization was for keeping Rocky's left jaw protected from a right hook, I knew Tommy was hooked on computers and software. For the remainder of our vacation, he talked about the power of software, and he began to think about how to write code for a computer game. He was no longer just interested in the façade of computer games, he wanted to know their programming innards.

Rocky Marciano took a dive on that day, but it was for a good cause---my son had discovered not only the fun of software programming, but its power as well. As mentioned, Tommy went on to study other programming languages during his college career---and I might add, making As in his classes. He worked for a while as a Web software designer.

Later Endeavors

Later, after Tommy became an expert in multimedia packages, he created several complex and elegant animated movies for my lectures about computer networks. In 1996, I made a two-hour TV presentation on the PBS Business Channel, and the show producers commented on the high-quality of Tom's visuals. My son was not altering the software of the multimedia programs (which cannot be done unless one has unique access privileges). He was manipulating input values to the software and controlling scores of buttons and keys on an Avid multimedia device. Tommy showed a flair for configuring and manipulating these very complex programs.

During the creation of the PBS program, I learned Tommy was the type of systems analyst and programmer who held the client's feet to the fire. He would not let me get away with a hazy statement about how an animation was to appear. His standard admonishment was, "Dad, you're being ambiguous. You haven't thought this through, have you?"

"Nope…thought you would do it for me."

"In your programming days, did you let your users off the hook so easily?"

"Okay, stop your badgering. Let me give it some thought, and I'll get back to you."

"Let's talk it through together, now."

I would liked to have had several Tommys on my project teams when I was involved in software development. Thanks to Sugar Ray and Rocky, Tommy became one of those programmers that we in the industry say are, "one in a hundred." I know. I saw his work on national television.

Postscript:

Here is an update to this story: Tommy left the world of programming, but he maintained and expanded his knowledge of the Web and social media. He now helps me get along in the Internet's apps. As of this writing, he has become a popular and respected musician on the West

Coast. He plays bass and guitar with Scott Weiland's new band, The Wildabouts, and is Scott's accompanying singer.

He has yet to get even with my Rocky Marciano joke. I'm keeping my guard up.

Addendum

I believe Rocky Marciano to be the best heavyweight boxer of all time, including Muhhahhamed Ali. At the risk of being KO'd by Ali fans, I even rank Ali behind Joe Louis.

The math backing my claim is simple: Unlike any other heavyweight, Rocky went undefeated in his career. He defended his title six times, against Jersey Joe Walcott, Roland La Starza, Ezzard Charles(twice), Don Cockell, and Archie Moore.

Proponents for ranking Ali ahead of Marciano state he had more competition. Really? More dangerous and hard-hitting than Jersey Joe and Archie? Consider this statistic: His knockout-to-win percentage of 87.75 is one of the highest in heavyweight boxing history. Ali's record was (61-56-5) with a KO percentage of 67.6 percent (He KO'd 37 opponents).

How about Joe Louis? His record was 69-/66/3 and KO'd 52 opponents, for a 75.3 percentage knock out rate.

To add fuel to the fire, I consider Sugar Ray Robinson to be the best fighter of *all* professional boxers. Just imagine: He went 85–0 as an amateur and 69 of the bouts were knockouts, with 40 coming in the first round. One can imagine opponents stepping into the right for the first round hoping they would still be around and stepping into the ring for the second round. Turning professional in 1940 (19 years of age), by 1951 his record was 128-1-2 with 84 K0s. (a 69.4 percent)

I consider Ali a wonderful fighter, a fine, kind man out of the ring, but an obnoxious man in the ring. I'm from the old school of boxing: You don't taunt your opponent, when your opponent has been felled by your fists and is lying semi-conscious on the mat.

If you contest my claims, put up your dukes!

