

Education Outside of School

I went to a conventional school but quite a lot of my education happened elsewhere. I ended up teaching in two different fields, economics and law, in neither of which I had ever taken a course.

WoW Ed

Suppose you are a comfortably well off parent. Almost everything your child wants you to buy him — toys, books, games — can be bought at what is, in terms of your income, a trivial cost. That makes it hard to do an honest and believable job of teaching your child the importance of saving, of deciding which things he really wants and which he can do without, skills that he will need, as an adult, to function in a world of limited resources.

If your child plays *World of Warcraft* he will learn the relevant lesson with no need for you to impose arbitrary limits. He will have a limited amount of gold and a considerable variety of things he would like to spend it on. Increasing that amount will require him to spend time doing daily quests, figuring out what he can craft and sell at a profit and crafting and selling it, perhaps, if he is a mage, running a magical taxi service creating portals to transport other characters for pay. Whatever his effort, he will probably not end up with enough gold to buy everything he wants.

The lesson works because it is, in its own odd way, real. These are the things he has to do in order to achieve the objectives he has himself chosen. The virtual world he is making these decisions in is in many ways like the real world in which he will make it. There are wealthy individuals, to whom things that seem expensive to him are cheap. There are beggars, people who hope to get money from others without giving anything in exchange. Some wealthy players are wealthy because they have found something that both makes money and is fun, such as auction house games of arbitrage, speculation, cartelization, and the like And he is doing all of it not because someone makes him, whether parent or teacher, but because he wants to. It is, in that respect, the ideal educational software.

That brings up another idea, one that WoW has not yet implemented but that it, or one of its rivals, could. Another real world skill that could be learned in a virtual world, perhaps better than in school, is language. I am imagining a version of World of Warcraft in which the non-player characters sometimes speak French instead of English. At the early levels it would be very simple French, with the meaning obvious either from parallels to English worlds or from context. As the game progressed, more and more French words would be used and understanding them would become increasingly important in playing the game.¹

Done right, the effect would be not that different from the way in which we learn our first language.

One commenter:

What I would also observe is that when these games are written with children's education in mind, they tend to lose value as educational tools. Children know when they are being patronised, and don't like being talked down to, which is what most children's computer games do.

¹ Although, so far as I know, nobody has done it yet, it should be possible to practice a language by setting WoW to that language and logging in on a foreign server where almost all players communicate in that language.

So I'm happy to accept a little adult content, and in return my kids get a real time boot camp in a lot of useful life skills.

How Kids Learn to Write Nowadays

Quite a lot of successful authors of fiction seem to have developed their skills not in college creative writing courses but in Dungeons and Dragons, as dungeon masters and/or players. A role playing game is, after all, an exercise in collaborative story telling. The things that make it work or not work are mostly the same things that make a novel work or not work. And, unlike a creative writing course, the participants are doing it because they want to, for the fun of it, not because someone else has told them that this is what they have to do to learn to write.

My own non-fiction writing largely developed during the years when I was producing a monthly column for *The New Guard*, a conservative student magazine on which I was the token libertarian columnist. There too, I was writing not as a school exercise but because I had things I wanted to say. My daughter, when she was younger, spent a good deal of time writing up and webbing battle reports describing, as if they were real, events she was part of in *World of Warcraft*; she is now a freelance editor, specializing in fantasy and science fiction. My son played, and still plays, historical simulation computer games, whose players sometimes produce After Action Reports, AARs, accounts of what happened in the game written as the work of a later historian.² He is now an aspiring novelist. I do not think either of them ever got into fanfic, the practice of writing stories set in the world of *Star Trek*, or *Harry Potter*, or *Lord of the Rings*, but a lot of other young people have. I will not be surprised to discover, ten or twenty years in the future, that some of the new generation of authors, especially fantasy authors, got their start there.

Which suggests to me that English classes in high school and college play a smaller role in teaching this generation how to write than their teachers might suppose.

The Great Games as Lessons

A very long time ago, back when long distance calls were expensive, I was in Europe trying to coordinate plans with my parents in America. It occurred to me that the problem I was facing, coordination with imperfect communication, was one I had met before — playing bridge. The two players that make up a partnership in bridge have identical interests, are for game theory purposes a single player, but are not permitted to talk to each other. They need to communicate in order to share the information needed to decide how high a bid they can risk making and what suit they should make it in, but they can communicate the contents of their hands only by the bids they make, which serve at the same time as moves in the game and an elaborate system of coded information.

It is sometimes tempting to make a bid that you think will get you to the right contract even though it misrepresents your cards, or to conclude that because your partner is too cautious you need to compensate by overstating the strength of your hand. There is sometimes a matching temptation to suspect that your partner's bid does not mean quite what you had agreed it would mean, and to modify what you do accordingly. That way lies ruin. The cardinal rule of bridge playing is "Trust your partner."

² For samples, see: <https://forum.paradoxplaza.com/forum/forums/aars-lps-and-fanfiction-general-discussions.224/>

It is a good rule for other games of coordination with imperfect communication as well. Such as [marriage](#).

Other games that have remained popular for long periods of time might be viewed similarly: chess as training for tactics, poker for bluffing and not being bluffed, *Diplomacy* for coalition building and its perils.

Computer Programming as a Children's Game

RobotWar is a computer game originally written for the PLATO computer system at Indiana University in the 1970's, then in 1980 ported to the Apple II and published commercially. Versions were later released for other computers.³ The player programs an onscreen robot tank, using a stack based language, RobotWrite, that looks to me like a simplified version of Assembly Language.⁴ A program can write to or read from registers, memory locations some of which control features of the tank or report things about it. A tank has a turret which is rotated by putting a number into a register. The turret has a gun and a radar.

A sample from Wikipedia, with comments, for a very simple robot:

```
SCAN
  AIM + 5 TO AIM          ; MOVE GUN
  AIM TO RADAR           ; SEND RADAR PULSE
LOOP
  IF RADAR < 0 GOSUB FIRE ; TEST RADAR
  GOTO SCAN
FIRE
  0 - RADAR TO SHOT      ; FIRE THE GUN
ENDSUB
```

The robot with this program sweeps its radar in a circle, firing off radar pulses, and when it detects another robot, fires a projectile set to explode at the correct distance as estimated by the radar pulse. This particular robot stands still throughout the entire battle, as it never assigns any number to its movement registers.

The first tank you program batters itself to death against the wall of the arena, so you add a subroutine to detect how far off the wall is by bouncing radar pulses off it and turn the tank around when getting too close. After a few iterations of the “load the program into the computer and watch what the tank does/see what was wrong/revise the program/load the program” cycle, you get a tank good enough to survive in the arena as long as nobody is shooting at it. You are now ready to put in a subroutine to rotate the turret and, if an enemy tank is detected, shoot at it. Since both tanks are moving, their relative position will be changing, so you need to figure out how your tank can deduce what direction the enemy tank is moving and how fast, so as to know where to look for the next shot. The enemy is shooting at you too, so you add a subroutine that detects damage by reading a register and randomly alters speed and direction in response.

³ The versions for Macintosh and Windows are called RoboWar. The Mac version does not run under the current operating system so would have to be run in Sheepshaver, which emulates OS9 on OSX machines, or some other emulator, or on a very old Mac.

⁴ I have seen it compared by others to BASIC or Forth. The version I encountered was RoboWar on the Mac — I don't know how close that is to the versions on other machines.

You have programmed your tank, your friend has programmed his, it is now time for a battle. Load both programs into the computer and watch your tanks fight. Each is controlled by its program, with no further input from the programmer. At each stage of improving your tank you make mistakes, create bugs, find them, eliminate them. You are learning to program for the fun of it — and, with luck, to keep your tank a little better than your friend's.

Silas Warner, the inventor of RobotWar, made computer programming, an intellectual skill sometimes taught to bright students in college and capable of leading to a well paying job, into a children's game.