A fantasy adventure game as a learning environment: Why learning to program is so difficult and what can be done about it.

Robert Moser
School of Computer Science and Engineering
University of New South Wales
moser@cse.unsw.edu.au

Abstract

A number of factors contribute to the difficulty of learning computer programming, and by examining some of these factors we can hope to design an environment that is more conducive to the learning process. I analyse some of these problems and find that one possible solution is to embed the educational content into a fantasy adventure game. Designing an educational game is, however, fraught with difficulties beyond those normally associated with writing such a program, as there are conflicts between educational and entertainment goals. Described are a few of these conflicts from the early stages of my project, and the compromises that they have engendered.

1.0 Why is Computer Programming So Hard?

Learning to program computers is a difficult process. Much like the process of acquiring any new language, becoming fluent in a modern programming language requires many different layers of skills built one atop another until the learner can piece letters together into words, words into sentences, and sentences into paragraphs or lines of a sonnet. On top of these difficulties add the fact that programming languages are artificial - designed to communicate in the unfamiliar realm of the computer rather than the natural world that we all grew up experiencing - and you are confronted by a daunting task indeed. Yet by trying to better understand some of the factors that contribute to this difficulty we can design a learning environment which will help the student along the road to understanding.

1.1 Programming is a multi-layered skill.

Programming isn't a single skill; it is a multi-layered hierarchy of skills, many layers of which will be active at the same time. Research on the learning patterns of students[1] shows that they acquire these skills from the bottom up, learning syntax before structure before style. During this process many bad habits can be learned on higher levels while the attention is focused on the lower.

1.2 It is unrelated to much of day-to-day experience.

The programmer's basic building blocks are, in their formal senses, alien to the computer novice. The learner is constrained to the bottom-up learning method described above by their lack of any previous knowledge, and the nearest parallels to programming for most people - language and mathematics - are full of misleading instances of negative transfer (a formal "or", for example, does not equate to the "or" of the average English speaker.)

1.3 It is learned in a single context.

According to schema theory we abstract from multiple examples an underlying structure that helps us to more easily store large amounts of information[9]. The skills of the programmer, however, are learned in a single context at a time; a constraint imposed by the necessity of learning the syntax of a specific language before any application of the skills can take place. This single context of learning makes it more difficult to derive schema for the use of those skills, while large increases in flexibility can be seen in students learning their second language and thereby working in a different context.

1.4 It is boring.

Syntax of programming languages is almost invariably obtuse, and for (at least partially valid) reasons that the beginner cannot hope to understand. In addition, most beginning subjects in programming are aimed at age groups, not skill levels. Students of the same age but of widely differing computer experience are alternately overwhelmed or bored by the same course.

1.5 It is intimidating.

Many people, particularly older generations and younger women, find computers in general and programming specifically to be a very intimidating area. These people will simply never try it, because they have labelled it as "too hard."

2.0 A proposed solution.

A proposed solution to the problem is to design a fantasy adventure game to teach basic programming skills. By working on a metaphorical level with a restricted programming language we allow the students to work with higher-level concepts before learning the details of a full general-purpose programming language. By setting the game in a completely different context from normal programming we encourage the abstraction of schema and situate the learning process in a more familiar environment. A one-on-one interaction with the computer allows the learner to progress at their own pace. Finally, a game is both non-threatening and fun, allowing us to take advantage of the known benefits of self-motivated learning.

The choice of a fantasy adventure as the solution to the these difficulties was not made by chance. In addition to helping to solve the problems inherent in the educational content, significant bodies of research in the areas of educational psychology, learning theory, and engagement support various aspects of the genre as appropriate for the task. Specifically, engagement[3,5], embedding the learning process into an

ITiCSE '97 Uppsala, Sweden
© 1997 ACM 0-89791-923-8/97/0006...$3.50

external environment[4], stimulating the student's curiosity[8,6], the vivid imagery of fantasy[2,10], and using intrinsic fantasy as educational rewards[8] have all been cited as aiding in the acquisition of new knowledge and skills. Finally, in the most practical sense, there is a direct mapping from programming computers to the concept of magic in a fantasy world.

3.0 Constructing the game; lessons from the trenches.

While the problems inimical to computer programming and the research in various fields seem to agree on the choice of a fantasy adventure game as a good potential learning environment, there is much less agreement when it comes to the actual structure and construction of that game. Where in the normal course of software design we would be concentrating on the user's goals, we find ourselves confronted with two different users with very different sets of priorities; the player desires to be entertained, but the instructor wants the student to learn. Different aspects of instructional and game design contradict as well as reinforce one another - compromises must be made.

3.1 Conflict: instructional vs. game design.

The primary influence on the construction of an educational game should be that of the instructional content itself. In this case we want to acquaint the student with a sample of some basic control structures common to many computer languages before introducing them to some higher-level concepts, so I chose to work with the following programming elements: assignments, boolean expressions, code blocks, conditionals, iterative loops, and conditional loops. Systems for designing classroom interactions can be partially utilised for the game to create ordered hierarchies of sub-skills for each of these skills, and lessons associated with each sub-skill which will scaffold the user through the learning process.

Unfortunately this very structured mapping of the learner's progress through the game conflicts with some of the basic tenets of engagement theory. Brenda Laurel tells us that the user's feeling of involvement is strongly tied to the range and impact of choices available to them[7]; by leading them through a predetermined sequence we reduce their choices to those of minimal consequence. It also greatly lessens the benefits of exploratory learning and exciting the curiosity of the user by removing a large portion of the unknown from the game. Yet the open, exploratory model suggested by these disciplines leaves no room for directing the learning process.

3.2 Compromise: a tiered structure.

Instead of giving up the benefits of either in exchange for the other, I chose to combine the two in something of a tiered structure; focal points in the learning process are identified and embedded in "funnelling" scenes in the plot, and these points act as gateways between freely explorable areas. We can thus assure that more difficult problems are not encountered until some scaffolding action has occurred, and it has the added benefit of allowing us direct the overall shape of the plot in a dramatically exciting and satisfactory manner while leaving the user their freedom of action.

The process of crafting all of this theory into an actual functional game structure turned out to be a cyclical one; from instructional content to plotline and back again. First, the initial focal points were found in the skill hierarchies. These tended to correspond to prerequisite knowledge needed to use the skill, and were mostly very simple (for example, learning about conditionals is impossible without some idea of basic comparisons.) Having determined what basic skills the learner should have acquired by this point, I designed a series of tests to check on their proficiency at these skills. These tests are then woven into the fabric of the plot(which so far consists of a fantasy world with the player as an inexperienced practitioner of magic) to form a consistent whole:

> *"An apprentice wizard, your master has left you on your own in his tower. He (or she) has left a magical guardian at the door who will not let you out until..."*

Perhaps the apprentice must demonstrate that he/she has completed their lessons for the day, or finished their chores, or done something nice for the guardian itself; these are the tests, embedded into the plot. From this point it is reasonably simple to describe the confines of the tower (which the user can freely explore) and to invent interactions that will help the user to learn the material tested for at the focal point - the guardian at the door. These interactions can be discovered and completed in any order, or even not at all, but so long as the learner has not yet demonstrated sufficient mastery they cannot progress beyond that point.

Now the process is repeated a second time, at a further remove. Having shown an understanding of prerequisite skills, the student is ready to be introduced to the next stages in the skill hierarchies. Again there is a new area to explore - the city outside the tower - and again there is a barrier to that exploration - a wall around the city, and a guarded gate. The gate is the next focal point, beyond which the design cycle can repeat still further, as necessary (I have found three repetitions sufficient so far.) Each of these funnels in the action acts as a mini-climax in an overall plot(e.g. finding and rescuing your lost master,) building tension towards the final dramatic conclusion.

4.0 Where now?

Having reconciled warring elements of theory in my design of the plotline, I now turn to the task of building the environment itself. Still in progress are the construction of a parser for my simplified programming language, the design of the game's interface, and the coding of the game engine itself. The resulting environment will be evaluated for transfer to programming so that the effectiveness of the technique can be determined.

References:

1. Bereiter, C. & Ng, E. "Three levels of goal orientation in learning." The Journal of the Learning Sciences 1 (3&4) 1991.

2. Bower, G.H., in L.W. Gregg(ed.), Cognition in Learning and Memory. New York: Wiley, 1972.

3. Bowman, R.F., Jr. "A Pac-Man Theory of Motivation: Tactical Implications for Classroom Instruction." Educational Technology 22(9) 1982.

4. Brown, J.S., Collins, A., & Duguid, P. "Situated Cognition and the Culture of Learning." Educational Researcher v.18 no.1 1989.

5. Celsi, Richard L., & Olson, Jerry C. "The Role of Involvement in Attention and Comprehension Processes." Journal of Consumer Research Sept. 1988.

6. Collins, A. & Stevens, A.L., "Goals and Strategies of Inquiry Teachers." in R. Glaser(ed.), Advances in Instructional Technology vol2. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1981.

7. Laurel, Brenda, Computers as Theater. Reading, MA: Addison-Wesley, 1991.

8. Malone, Thomas W., "Toward a Theory of Intrinsically Motivating Instruction." Cognitive Science 4 1981.

9. Norman, D. A. & Rumelhart, D. E. "Representation in memory." in R. C. Atkinson, J. J. Herrnstein, G. Lindzey, & R. D. Luce (Eds.) Handbook of Experimental Psychology. New York: Wiley, 1988.

10. Paivio, A., Imagery and verbal processes. New York: Holt, Rinehart, & Winston, 1971.