What is an ILM?

An Interactive Learning Module is a software entity which helps students to understand a concept by interacting with the software. There are several excellent examples of ILMs (for the math discipline) at eNLVM.

Characteristics include:
1. Educational. Needs to teach a computer science concept. The module should be usable in teaching basic concepts OR designed for a specific CS class (like data structures, advanced algorithms, etc).
2. Written as a Java Applet (for consistency). Flash or JavaScript could also be used.
3. Allow user to interact with software in meaningful way.
4. Can be used in a variety of lessons.
5. Compelling – something you envision students wanting to experiment with. The idea is that material will be taught more effectively by emphasizing problem solving.

Why ILMs?

Rather than have students create “throwaway” software, this semester we are creating software which will become part of the materials utilized by the funded NSF curriculum grant. The website is at http://csilm.cs.usu.edu

Example ILMs

1. The balls on the left are to be exchanged with the balls on the right by a sequence of moves. Any ball can move into an adjacent empty slot. Any ball can jump over a single neighbor to an empty slot. Students interact with the module by dragging the balls, identifying (and encoding) a strategy, and observing how the number of steps grows with the problem size. This same basic Interactive would be used in multiple eObjectives. In an introductory course, the exercise demonstrates strategy and planning needed to solve problems. As an exposure to algorithmic thinking, we consider the ability to write an algorithm to solve the problem for a varying number of balls. Abstraction is introduced as we construct rules which are general purpose. Algorithm complexity and analysis are the focus as we seek to evaluate the optimality of our solution.

2. The graphical interface for another proposed Interactive is shown below. The students are asked to create a cipher for a text of their choosing. They will experiment with cracking codes by sending their cipher text to others in the classroom.
3. The sort detective shown to the left is an example Interactive that is used in our data structures course. Working in small groups, students are asked to identify which sort each button calls based on how much work is done, how many times the data is moved, and whether or not data is kept in the original order when the keys are identical. This Interactive is enormously effective in teaching sorting concepts like stability and obliviousness. We asked typical final exam questions about complexity, such as “You have two algorithms which are of the same complexity, yet one is much faster than the other on the test data. Give two reasons to explain how this could happen.” In previous semesters, this has been a challenging question. After introduction of the ILMs, many students confidently stated, “Well, it could depend of the particular input data. In sorting, for example, if data is mostly sorted, some sorts take advantage of that initial order while others do not.”