A Game-Centered Approach to Teaching Computer Programming

Donald House, Cory Buckley, Chris Malloy, Zack Inks and Brian A. Malloy
School of Computing
Clemson University
{dhouse, malloy}@cs.clemson.edu

The art of computer programming requires the programmer to master the rigorous syntax and semantics of a programming language, and to understand the computational architecture underlying the language. The project that we describe centers around the development and use of an architecture simulator that provides the student programmer a view into the computer, and how data and control flow are coordinated as a computational process executes.

The above figure depicts the initial interface. As a student’s program executes, the animation illustrates both control flow and the movement of data between input, memory, registers, and output. When the student masters concepts at this level, the machine complexity increases, and new problem sets are presented that require a more advanced architecture, including memory laid out to handle a dynamic heap and call stack, as well as global and static variables.

To motivate the development of programming skills, we incorporate the programming tasks into the context of a video game, which forms the focus of our presentation. The game that we present is a role-playing game, written in Unity 3D, where the player assumes the role of a character that lives inside of a computer and must struggle against computer viruses, robots and other enemies. To overcome the enemies, the player must solve puzzles that are generated by the enemies, or that populate the game as an artifact of enemy activity. The video game and the programming tasks are interwoven so that completion of a programming task facilitates progression through the game and progression through the game earns clues and other artifacts that facilitate progression through the programming puzzles.