SIGCSE 2007 Faculty Poster Submission

Using Blender and the Blender Gamekit with Python in CS2

Adrian German[1]
Department of Computer Science
School of Informatics
Indiana University Bloomington

dgerman@indiana.edu
(812) 855-7860 (TEL)
(812) 855-4829 (FAX)

Tradition says that "[a]n introductory course (and its successor) in programming [is] concerned with three aspects of programming: 1) How to solve problems, 2) How to describe an algorithmic solution to a problem, and 3) How to verify that an algorithm is correct." The computing landscape has changed significantly since this recommendation was first made (Gries, 1974) and with each major change instructors have had to meet the challenge of applying these guiding principles to integrating new technologies and methodologies (OOP, GUIs and event-driven programming, the web) into the classroom.

For about 8 years (1997-2005) the introductory programming sequence for non-majors at Indiana University (A201-A202) has been taught in Java. Last Spring we have switched to Alice and Python in A201 and this Fall we have introduced the Blender Gamekit in A202 (along with Python, Pygame and livewires). A202 is now[2] listed as a "survey of modern approaches to programming techniques with an emphasis on developing real-time 2D and 3D interactive environments for visualization, simulation, education or entertainment".

Our poster reports on the first administration of this new course (currently in progress): goals, means, results, student perceptions, difficulties encountered and ways in which we managed to overcome them, as well as conclusions for the future. We had three goals in mind when we restructured the course:

a) dispel the myth that computers are intrinsically hard to use and that computer programming classes always require an inordinate amount of effort for achieving even the slightest meaningful goal; Blender provides a powerful interface that is accessible to people with limited programming skills, yet models can also be generated through a program. The tool is powerful and results are impressive (as we hope our poster will show). Even more important, Blender is extensible. All extensions available are through programming languages such as Python.

b) expose students to a world-class 3D modeling and animation tool; the hope is that by being exposed to a good tool more students would be attracted to the field that builds these tools (the class is for non-majors). The visual dimension of using Blender provides increased motivation. Blender is free and open-source.

[1] http://www.cs.indiana.edu/People/auto/d/dgerman.html
[2] http://www.cs.indiana.edu/classes/a202

c) teach students how they can use their (newly acquired) A201 knowledge in implementing moderately complex projects. For any kind of meaningful project one needs an API. Relying on, and interacting with, an existing library/module is essential for any programmer, and in this class we're extending Blender's capability by writing Python code in a variety of ways. The Blender game engine is a "true" engine, in the sense that each object in the game has its own logic, sensors, actuators; there is no higher intelligence that controls the game. (Almost the same is true of Pygame/livewires). Thus designing a game using such an engine is essentially an event-driven programming project, very similar to developing a GUI based program with widgets, events, listeners and so on. It’s not only the visual aspect that motivates students to apply themselves more to the study of programming in such a context; it’s also the ability to add interactivity to a scene/model. And by designing and implementing their own ideas students achieve all three objectives listed at the beginning: solve problems, come up with an algorithm, prove that it’s correct. But even more important, they are able to reach a meta-level that was not accessible before: they choose their own problems. By choosing problems students implicitly communicate more about their goals, motives and personality to their instructors and this in turn creates a more intense pedagogical experience for all involved.

We include a selection of snapshots representative of the examples and projects currently covered in class; they’re all being considered for inclusion in the poster.