Workshop Title: **Teaching Networked (Real-Time) Multiplayer Game Design in CS1/CS2**

Presenters:

Adrian German (Contact Person)
Department of Computer Science
School of Informatics
Indiana University
Bloomington, IN 47405, USA
Phone: 812 855-7860
Fax: 812 855-4829
dgerman@indiana.edu

Bjorn Ottesen
Department of Telecommunications
Indiana University Bloomington
bottesen@indiana.edu

Abstract: Network programming, traditionally considered an advanced topic is usually approached with trepidation and fear by both instructors and students. But languages and tools developed in the last decade have completely taken the gnarl out of a topic that (as we show) can now be easily taught early in CS2 or even late in CS1. Our hands-on tutorial workshop presents a comparative development of a basic real-time networked multiplayer game in Python and in Java. The result (http://www.cs.indiana.edu/~dgerman/tutorials/sigcse2008.html) is a simple and scalable pedagogical framework, whose outcome is always invigorating and motivational.

Intended audience: Both new and experienced CS1/CS2 users of Python, Java.

Presenter Biographies:

Adrian German is a lecturer in Computer Science with Indiana University Bloomington where he has taught introductory programming classes (Python, Java, Perl, Ruby) and advanced web administration and programming classes for the last ten years. Has twice received department's teaching excellence award (1999 and 2003) and has been nominated four times for IU Students' Choice award and twice for the IU Trustees' Teaching Excellence award. He has authored a total of four books: "sOAKed in Java" with McGraw-Hill, "Soliloquy (in Java)", "The (mostly) Open-Source Web" (both with John Wiley), and most recently "The Essentials of Web Programming" with Pearson Education; all four books are with the custom publishing departments, but an extension of the latest is currently in development with O'Reilly.

Bjorn Ottesen is a graduate student at Indiana University pursuing a Masters' in Immersive Mediated Environments. He graduated with Honors in 2006 with a Major in Informatics specializing in media production. In addition Bjorn has throughout his academic career focused on mastering the art of programming with languages such as Java, Python, and most recently Actionscript for Flash (http://www.masterbjorn.com).

Materials provided: Each participant receives a CD with the software to be installed on his/her laptop, along with installation instructions. The CD will also include copies of all the materials (in both html and pdf format) that will be distributed in paper/handout format for the workshop.

Audio/Visual and Computer requirements: Ideally participants will have internet access and laptop power at each seat. We will also need a video projector, for presenters to guide step by step the development during the workshop.

Laptop Required: all participants will need a laptop to develop and then take part in the final multiplayer game; see below for operating systems considerations.

Space and Enrollment restrictions: No restrictions of any kind, however an enrollment of about 15-20 would probably be ideal.

Other critical information: The methodology presented in this workshop was initially developed for a networked game design class taught in spring '02 and then repeated in the spring of '03. Projects in the class were presented at the annual MIME student festival. Theoretical extensions of this framework have been presented at ICPADS '04 (in Los Angeles) and FIE '04 (Savannah, GA). The Java component of this workshop has been successfully included in 300 and 200 level CSCI non-major classes, as well in successive JETT workshops held at Indiana

University in '04 and '05; we have revised the materials based on the feedback received from colleagues/students, and confidently guarantee a smooth experience for SIGCSE participants.

It would be worth pointing out that we have designed the workshop in two parts, of 90 minutes each: first the Java development, then the Python exercise. The first part uses RMI and works on Macs, Linux and Windows; no specific game engine is used. The Python section uses Pygame and livewires as a game engine and plain sockets for networking; this part requires a Microsoft Windows platform. Overall the network is entirely transparent using Java RMI, and almost transparent in the Python approach. Common to both approaches is the ability to serialize and de-serialize complex structures automatically when sending them back and forth over the network, which eliminates the need to parse, a mechanism that is very similar to what SOAP/XML offers; but while SOAP/XML is way too general and complex to even hope to discuss it at the introductory levels we are aiming for, the methods we use in this workshop are entirely operational at those levels and simply work like a charm.