Nifty Assignments

Nick Parlante, Julie Zelenski
(moderators)
Stanford University
nick.parlante@cs.stanford.edu
zelenski@cs.stanford.edu

David J. Malan
Harvard University
malan@post.harvard.edu

Mark Sherriff
University of Virginia
sherriff@virginia.edu

Thomas P. Murtagh
Williams College
tom@cs.williams.edu

Daniel Zingaro
University of Toronto
daniel.zingaro@utoronto.ca

Zachary Dodds,
Wynn Vonnegut
Harvey Mudd College
dodds@cs.hmc.edu
wvonnegut@hmc.edu

Todd W. Neller
Gettysburg College
tneller@gettysburg.edu

Categories and Subject Descriptors
K.3.0 [**Computers and Education**]: General.

General Terms
Algorithms, Design, Languages.

Keywords
Education, assignment, homework, project, repository, library, nifty, object oriented programming.

Introduction
Though we dream of awesome lectures, much of what students actually take away from a course are due to the assignments. Unfortunately, creating successful assignments is time consuming and error prone. With that in mind, the Nifty Assignments session is about promoting and sharing successful assignment ideas, and more importantly, making the materials available on the web.

Each presenter will introduce their assignment, give a quick demo, and describe its niche in the curriculum and its strengths and weaknesses. The presentations (and the descriptions below) introduce each assignment. The Nifty Assignments home page, **http://nifty.stanford.edu** provides more information for each assignment, with handouts, data files, starter code, etc. Nick's running ACM editorial: given its mission to promote CS, the ACM should distribute all its materials freely on the web.

If you have an assignment that works well and would be of interest to the CSE community, please consider applying to present at Nifty Assignments at: **http://nifty.stanford.edu**

Picobot: CS 1, Assignment 1, Problem 1 - (CS1) Zachary Dodds, Wynn Vonnegut
YAK!? That is, Yet-Another-Karel!? Surely such a thing is redundant in 2010. This assignment is why we don't think so. Picobot is a Karel-like assignment suitable for the first problem of the first assignment of a first CS course. We have used it that way for the past five years. We have also used it as a hands-on introduction to "What is CS?" for visiting applicants, orientation participants, and in high-school outreach. Beyond the accessibility that all Karel-like assignments boast, Picobot has idiosyncrasies that suit its position as CS1-Hw1-Problem1 well:

- It is conceptually small and self-contained, needing no software beyond a browser.
- It is language-independent: it does not use Python, C, C++, Java, Scheme, or any other language.
- Because of this language-independence, Picobot is background-independent: every student successfully completes the first environment, and no student has completed the most difficult environment.
- Because of this background-independence, it reduces the "show-off" factor that we have found can creep into early CS 1 lectures among certain students.
- It asks students to program, but also concretely motivates CS beyond programming:

  Applications: It is a discretized version of the environmental coverage problem in robotics and path planning.

  Complexity: The number of Picobot rules and states are tangible examples of computational complexity, and we challenge students to create maximally efficient solutions.

  Computability: Not all environments are computable! We need not look elsewhere to motivate fundamental issues of (un)computability.

- Perhaps most importantly, Picobot is simple enough that we can assign it as a final project in CS 1. With that project, students realize they can now *build* the simulator that had introduced them to CS at the start of the term!

The Game of Pig - (CS1) Todd Neller
The Game of Pig is a folk jeopardy dice game that offers one of the best fun-to-SLOC (source lines of code) ratios of any game. One would be hard pressed to find another game with such simple rules that engages and leaves one on the knife-edge of indecision.

SIGCSE'10, March 10-13, 2010, Milwaukee, Wisconsin, USA.
ACM 978-1-60558-885-8/10/03.

As such, it is a teaching treasure. Mathematics instructors have long used this game to teach concepts of probability. We here offer a rich collection of CS1 Pig exercises that illustrate bottom-up development, Monte Carlo simulation, object-oriented design, and simple GUI design.

The rules of Pig are simple: Two players race to reach 100 points. Each turn, a player repeatedly rolls a die until either a 1 ("pig") is rolled or the player *holds* and scores the sum of the rolls (i.e. the *turn total*). At any time during a player's turn, the player is faced with two decisions:

- *roll* - If the player rolls a

  1: the player scores nothing and it becomes the opponent's turn.

  2 - 6: the roll is added to the player's turn total and the player's turn continues.

- *hold* - The turn total is added to the player's score and it becomes the opponent's turn.

Sounding Off: Digital Sound Processing (CS1) - Daniel Zingaro

The yearly SIGCSE Nifty Assignments session almost invariably includes assignments based on images or animations. To give more non-visual learners the opportunity to exhibit their strengths, I propose a CS1 assignment that focuses on digital sound rather than digital images.

In this assignment, students begin by implementing simple filters on sounds: reversing sounds, adding echo, mixing sounds together, and changing volumes. These filters can each be written in ten or fifteen lines of code, using objects that represent sounds and their samples. Students experiment with the filters on standard wav files, helping them appreciate the assignment's practical context of digital sound processing. With this emerging understanding, students use their filters in a program that generates songs from "notestrings" — strings of data representing the sequencing of notes. Notestrings encode features such as note length, volume, octaves, and the ability for multiple "hands" to play simultaneously. I provide sample notestrings of varying complexity so that the students hear recognizable tunes when they have properly generated the corresponding songs.

The assignment gives students practice with loops, array/list processing, string manipulation, and function composition. It can be extended by encouraging students to implement more filters or more notestring features. Finally, though it appeals to a domain with which most students are familiar, it particularly engages musically-inclined students who may generate notestrings to share with the class.

CSI: Computer Science Investigation - (CS0-CS2) David J. Malan

Each semester, I somehow manage "accidentally" to format my digital camera's CompactFlash (CF) card on which are (were!) some terribly important photographs. And so I make for our students a "forensic image" of that CF card (using a free program called dd) and beg them to recover my photos. They proceed to write a program that iterates over that image's bytes, searching for tell-tale signs of my JPEGs, recovering each photo in turn.

By assignment's end, most students are amazed that such an important process reduces to some file I/O, one or two loops, and familiarity with hex. We daresay that students are excited that, with just a few weeks of CS under their belt, they can actually tackle an actual, real-world problem (that may have even happened to them).

We assume for this problem set familiarity only with loops. File I/O and hexadecimal notation are introduced in the problem set's own narrative.

Upon recovering my photos, students learn that each is of some "non-obvious but identifiable" location on campus. They are then challenged to find and photograph themselves in front of each of those spots in exchange for what we promise will be an amazing prize.

Encryption Chase (CS1-CS2) - Mark Sherriff

Do you or your students enjoy scavenger or puzzle hunts? The Encryption Chase gets CS1 or CS2 students working through the basics of encryption in an active-learning environment through a puzzle hunt where the puzzles are all based around various encryption schemes. Students work through clues around your building or around campus using Caesar ciphers, Vigenere ciphers, RSA, and other encryption schemes. The chase can be as rigorous as you want, based on the difficulty of the clues you present. There can be as little or as much programming involved as you like, so it's possible to tailor this assignment for both CS1 and CS2 students. After a successful chase, students come back with an appreciation of the various forms of encryption that have been used throughout history. An example chase will be presented with some tips and tricks on how to write your own custom chase for you students. Come to hear about the assignment yourself, and you may find yourself in the middle of your own chase!

Chatting AIM-lessly: Implementing an IM Client (CS1-CS2) - Thomas P. Murtagh

Our students spend much of their time (including class time!) texting, tweeting, or chatting using AOL or Google. Why not take advantage of this enthusiasm for short strings?

In this assignment, students construct a program that can be used to chat using AOL's IM messaging system. The completed program allows its user to conduct simultaneous chats in separate windows with several individuals and to keep track of which of their friends are online. The assignment only requires support for plain text messages, but enthusiastic students have often implemented extensions for emoticons, fancy fonts, etc. Their enthusiasm is partly inspired by the fact that they can use their own programs to chat with others who are using *real* IM clients.

The assignment is designed to be used about half way through a CS1 course. It is divided into two one-week exercises. The primary goals during the first week are to give students experience writing loops and manipulating strings. The goal of the second week is to introduce the use of multiple classes. An instructor could choose to either use the complete assignment or just the first week. The program completed in the first week is fully functional, but all chats are displayed together in a single window.

The assignment was designed for a course that uses Java and depends on a supplied class that provides simple mechanisms for exchanging packets with AOL's TOC server.