New Orleans JAZZED about Engineering Education

Paper ID #15269

WORK IN PROGRESS: Teaching Broadly-Applicable STEM Skills to High School Sophomores Using Linux and Smartphones

Prof. Daniel Brian Limbrick, North Carolina A&T State University

Dr. Daniel Limbrick is an assistant professor in the Electrical and Computer Engineering Department at North Carolina Agricultural and Technical State University (NC A&T). As director of the Automated Design for Emerging Process Technologies (ADEPT) laboratory at NC A&T, he researches ways to make computers more reliable (i.e., radiation hardening) and scalable (e.g., three-dimensional integration) through novel approaches to electronic design automation (e.g., Logic Synthesis, Placement, Routing) and computer architecture. Dr. Limbrick also researches ways to design digital microfluidics-based lab-on-a-chip to be faster and capable of more complex functionality.

Dr. Limbrick teaches computer design at multiple levels of abstraction. He created the STEM Scholars program at NC A&T, which trains high school students in Linux shell scripting, data plotting, and algorithm design and engages them in an algorithm design competition. Additionally, Dr. Limbrick created a Linux laboratory for undergraduate students at NC A&T, with the goal of exposing students to Linux, shell scripting, open source software, and Linux-specific design tools in order to prepare them for future employment in engineering.

Prior to his time at NC A&T, Dr. Limbrick held a postdoctoral appointment at the Georgia Institute of Technology and received his graduate degrees (M.S. 2009, Ph. D. 2012) in Electrical Engineering at Vanderbilt University.

Work-In-Progress: Teaching Broadly-Applicable STEM Skills to High School Sophomores Using Linux and Smartphones

Introduction

The projected workforce demand in science, technology, engineering, and math (STEM) fields in the United States is currently greater than the projected supply of STEM workers. Additionally, African-Americans continue to be underrepresented in STEM fields. The STEM Scholars Program aims to increase the number of African-Americans in STEM fields by preparing high school students for the STEM college curriculum.

Many first-time students enter college without any background knowledge in their chosen STEM major. This lack of exposure puts them at a disadvantage, as they are learning course material concurrently with basic STEM-related skills, such as computing and algorithmic thinking. This program aims to train high school students in these skills to increase their interest and aptitude in STEM fields. As a result of this effort, students will be more prepared for STEM majors and be more attractive to STEM programs.

STEM Scholars is an academic year-long program composed of bi-weekly hour-long seminars that teach students through hands-on training of Ubuntu Linux, bash shell programming/scripting, and GNUplot data plotting software. Additionally, the students compete in teams to design an algorithm that solves a puzzle in Flow Free, a puzzle game that runs on Android/iOS platforms. This program aims to engage high school students by using example problems that are relevant to their current studies. Therefore, math problems were taken from a Scholastic Aptitude Test (SAT) workbook as well as a Calculus textbook. An additional goal is that the students continue the exercises independent of the seminar. To this end, the puzzle game can be accessed from a smart phone and the tools that are used in the seminar are freely available.

Flow Free Background

Flow Free is a puzzle game that runs on Android/iOS platforms. The game board is composed of a grid and colored dots, and the user has to connect the same colored dots together without overlapping other lines, and using up ALL the free spaces in the board. This problem (shown in Figure 1) is similar to classic puzzles that involve connecting similar dots.[1,2]

The problem also has relevance in the field of integrated circuit design. One of the design challenges of building integrated circuits is connecting the components together. Researchers are continuously trying to find ways to connect modern devices in a given space with wires. The length of the wire must be minimized to reduce power and delay. Additionally, unrelated wires cannot intersect in order to prevent shorted signals. By exposing the high school student to Flow Free and challenging them to approach the problem scientifically, they are being exposed to a relevant engineering research area.

Learning Linux

Throughout the course of the program, the students are taught Ubuntu Linux. The objective of exposing high school students to Linux directly addresses the growing influence of computational science, high performance computing, and big data analytics on all STEM fields. Linux is a popular platform for simulation tools for computational biology,[3] and physics.[4] Additionally, Linux

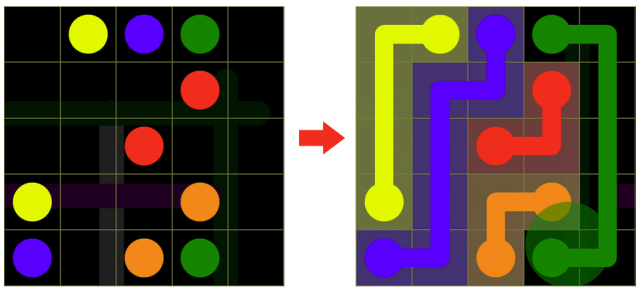

Figure 1: Example puzzle from the game Flow Free. Each pair of dots with the same color need to connect without overlapping. There are hundreds of freely available puzzles for this game. Ten puzzles will be chosen at random to engage the students in competition.

is at the heart of embedded systems.[5] Ubuntu Linux was chosen because this distribution has a program interface (i.e. Unity) that is most similar to Windows, the most popular operating system. The similarities might ease the adoption of this operating system by the participants. Additionally, Ubuntu is one of the most popular Linux distributions. This section provides details to the activities that expose the participants to Linux.

Competition Component: Emulating competition environment

The students were given tutorials on all programs needed to prepare for the competition (e.g., GNUPlot, Dia, Bash). Additionally, the students were taught how to open VirtualBox, a hardware emulation tool, to run an Android-based cellular phone. This emulated environment was used to run the Flow Free game so the students could share a screen while they devise strategies to solve puzzles.

Interactive Exercise: Bash tutorial

To help understand how the linux command-line interpreter works, the students were given a tutorial that showed them how to execute arithmetic operations on variables. The Bourne Again Shell (Bash) was chosen to because it is the default shell of Ubuntu. The arithmetic operators detailed in the tutorial included add, subtract, multiply, division, and modulo.

Standardized Test Preparation: Numbers and sets

Additionally, the tutorial introduced the *sort* and *comm* commands to show the students how to deal with numerical sets. The students engaged in an exercise where they found the intersection of two sets. The students were encouraged to extend this tutorial to cover SAT/ACT problems involving numerical sets.

Flow Chart

In order to teach algorithms without the more difficult task of teaching a programming language, the students are taught how to develop a flow chart. This section provides details to the activities relating to flow charts.

Competition Component: Describing the Algorithm

On the competition day, the students will present flowcharts that they created in the program, Dia, a diagram creation program. These flowcharts will be followed exactly to see if it is capable of solving ten puzzles.

Interactive Exercise: Blindfold and Lead

In order to teach the students the value of giving explicit instructions the students were required to navigate their peers across the room. One student volunteered to be blindfolded and two students volunteered to give instructions. The blindfolded student had the objective of traveling from one side of the room to the other side based purely on the instructions of his/her peers. This exercise emphasized the level of precision necessary to describe a sequence of steps.

Standardized Test Preparation: System of Equations

The students were given a pair of linear equations and instructed to solve for the unknown variables using an algorithmic approach. The approach was required to be documented using a flow chart. After the exercise, the students were encouraged to use this algorithm on SAT/ACT problems involving systems of equations.

Graph Visualization

A necessary skill for students majoring in STEM is the ability to plot data. To introduce this skill, the students are taught GNUPlot. GNUPlot was chosen because it is an open-source implementation of a command-line plotting software, it connects directly to other useful academic software such as Octave, and there are extensive tutorials available on the internet. This section provides details to the activities relating to GNUPlot.

Competition Component: Drawing the Board

As part of the presentation that the students will give on the day of the competition, they will be required to show a plot of a puzzle scenario using GNUPlot. This plot will contain labeled coordinates for each dot. The students will write a description of how their flow chart would interpret this figure.

Interactive Exercise: Position and Direction

The students were asked to stand in two rows, forming walls. Six students volunteered to stand at specific locations between these walls. Their position was meant to mimic the colored dots from Flow Free. Another student volunteered to solve the puzzle blindly (i.e., without looking at the other students). The information for the position of the students was given to the "blind" student through an oral description by the other students. After solving the puzzle, the students discussed what they learned about defining a problem for a computer to understand (e.g., specifying positions relatively).

Standardized Test Preparation: Shaded Regions

As part of a tutorial on GNUPlot, the students learned how to plot mathematical functions and visually show the difference between these functions using shaded regions. The students were encouraged to use this approach on shaded region problems on the SAT/ACT exam.

Competition

The students will use the knowledge they acquired (e.g., algorithms, technical presentation) from the bi-weekly seminar to develop an algorithm to solve the puzzles in the game. The algorithms will consist of a series of written steps to solve the puzzle without knowing the layout of the puzzle in advance. The students will compete to create the best algorithm based on the following criteria: (1) the highest number of puzzles solved, (2) the fewest number of steps in the algorithm, and (3) the best presentation of the algorithm. The team with the best overall algorithm will receive a $25 gift certificate for each team member.

Assessment

Success of the program will be measured through the results of a post-assessment survey (quantitative 5-point Likert scale) of the participants, as well as a survey of non-participants from higher grade levels of the same high school. The measured outcomes will be based on the participants' interest, knowledge, and future engagement in algorithms, research, Linux, and pursing STEM careers. Data on one year of the program will be available to present at the conference.

Additionally, the principal of the high school that is involved with this program has agreed to provide SAT/ACT scores and future graduation data (e.g., college major) of the participants as well as of non-participant classes. This information will be used to measure the effectiveness of the program as well as the improvement year-to-year and the improvement cohort-to-cohort.

References

[1] H. E. Dudeney and M. H. Dudeney, *Puzzles and curious problems*. T. Nelson and sons, ltd., 1932.

[2] A. B. Adcock, E. D. Demaine, M. L. Demaine, M. P. O'Brien, Reidl, F. S. Villaamil, and B. D. Sullivan, "Zig-Zag Numberlink is NP-Complete," *Journal of Information Processing*, vol. 23, no. 3, pp. 239–245, 2015.

[3] M. Levitt, "The birth of computational structural biology," *Nature Structural Biology*, vol. 8, pp. 392–393, August 2001.

[4] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, "Mdanalysis: A toolkit for the analysis of molecular dynamics simulations," *Journal of Computational Chemistry*, vol. 32, no. 10, pp. 2319–2327, 2011.

[5] I. McLoughlin and A. Aendenroomer, "Linux as a teaching aid for embedded systems," in *2007 International Conference on Parallel and Distributed Systems*, vol. 2, pp. 1–8, Dec 2007.