



Voice

CS FOR EVERYONE

The Voice of K-12 Computer Science Education and its Educators

Volume 3, Issue 2

September 2007

Inside This Issue

FEATURES

*Annual Symposium
Connecting With CS
Make It Fun!
AccessComputing
Israel Improves CS
Education*

COLUMNS

*Digi-Know
Classroom Tools
Spotlight
Our Story
Bits & Bytes*

INFO BRIEFS

*Volunteer Power
Staff
Contact Info
Contribute to CSTA
The Advocate Blog
Meet the Authors
Mark Your Calendar
Resources*

IN THE NEXT
ISSUE OF
THE VOICE

Your CSTA
Membership

Annual Symposium Offers New Perspectives

Making CS Relevant

Kate Conley, Periodicals Director, ISTE

"That's cool! ... It's so much easier." Although Technology Educator, Miguel Marquez, from Newton Country Day School in Newton, Massachusetts, said this during the one-hour Python session, the sentiment could easily have applied to the whole Computer Science & Information Technology Symposium. Held June 28, 2007, in Atlanta, the 8th annual symposium offered 22 sessions focused on hot topics for high school computer science (CS) and information technology (IT) teachers.

Nearly 150 attendees came from as near-by as Atlanta and as far away as Brazil to learn about the Alice, Visual Basic, and Python programming languages, teaching 2D arrays in Java, Web and Flash programming, wikis, GIS, trouble-shooting the lab, and how to broaden participation in urban-school computing classes just to mention a few of the sessions. Co-hosted by the Computer Science Teachers Association and the International Society for Technology in Education's SIGCT, the full-day program is carefully designed to help CS, IT, and computing applications educators navigate the complex and creative environment of computing education today.

In the opening session, SpelBots: Using Robotics to Inspire CS and IT Learning, Dr. Andrew Williams, associate professor of computer and information science and advisor to the SpelBots team at Spelman College, and two of his students demonstrated how robotics are being used to engage women in CS courses. Sophomore Whitney O'Banner and Senior Andrea

Roberson brought their robot dogs and had them perform in choreographed unison to "Who Let the Dogs Out," much to the delight of the audience. Williams said it's about "getting students to be creators not just users of technology." The students are part of an all-female, all-Black team competing in RoboCup, an international robotics competition. Williams sees teaching robotics as a way to promote creativity and innovation and to increase participation by women in the field of computing. "Money alone isn't a good motivator for women. Seeing a role model, a Black woman, doing it is what got me interested in CS," said O'Banner.

Repeat attendee Ann Blocksom, an AP CS teacher from Stone Bridge High School in Reston, Virginia, said in her evaluation, "I really look forward to these symposiums. As a CS teacher, most staff development is not relevant to my discipline. This is so great because everyone here is dealing with the same issues. I always go home with new ideas and [feel] re-energized for teaching. The effort and planning is so evident. It's like Christmas for CS teachers!"

Microsoft has been generously supporting the CS & IT Symposium for the past seven years, and organizers say they are already seeking funding for a symposium next year in San Antonio, Texas.

For more details on the symposium or to download presenter materials, visit www.csitsymposium.org/. For information regarding sponsorship, contact Chris Stephenson at cstephenson@csta.acm.org.

Volunteer Power

CSTA wishes to thank:

Joe Kmoch

for his dedicated work to launch the Source Web Repository

Bill Madden

for sharing his vision for the ACM Model Curriculum for K-12 CS Level III course

Staff

Chris Stephenson

CSTA Executive Director

Phone: 1-800-401-1799

Fax: 1-541-687-1840

cstephenson@csta.acm.org

Pat Phillips

Editor

Phone: 1-608-436-3050

Fax: 1-928-505-5220

cstapubs@csta.acm.org

Executive Officers

Robb Cutler

President

robbc@harker.org

Michelle Hutton

Vice President

mfh@pobox.com

Committees

Curriculum

cstacurriculum@csta.acm.org

Membership

cstahelp@csta.acm.org

Professional Development

cstapd@csta.acm.org

Research

cstaresearch@csta.acm.org

Certification

cstacertification@csta.acm.org

International

cstainternational@csta.acm.org

Grants

cstagrants@csta.acm.org

Connecting with Computer Science

An Interdisciplinary Approach

Anne Condon

Many students who are passionate about art or music, or who are inspired by goals of improving health care or the environment, don't see the relevance of computer science (CS) courses. This misperception has contributed to the underrepresentation of women and minorities in technology careers and is unfortunate since, in today's world, a command of technology gives one an edge in any endeavor.

The Computer Science Department at the University of British Columbia provides several programs for students who wish to combine their interests in other fields with an education in CS. The percentage of women in these programs is significantly higher than in the core CS major program. At the introductory level, the CS 101 class provides an interdisciplinary view of CS. I initiated this course in 2004, and continue to develop it with colleagues Holger Hoos and Steve Wolfman.

Our goal is to convey fundamental concepts in CS by situating them in domains which are relevant to the students. The course does not aim to be comprehensive in covering core CS concepts; if a student finds just one course module to be compelling, the course has been successful. Hands-on exercises and lab activities are integral to each module. The course is not labeled as being for non-majors, but rather encourages students in the class to continue on to other CS courses. The course is cross-listed with Women's Studies, since the approach is

inspired by feminist research on pedagogy, and the cross-listing attracts women to the course.

A number of fundamental CS concepts are reinforced in multiple modules. For example, digital representation of data is exemplified by the genetic code project in the biology module, as well as in the representation of images and music in the art module. Students are encouraged to think critically about user-interface design issues when using tools such as drawing programs or a genome database. This perspective helps students understand that the difficulty in using computer tools is not always their fault, and that, through the field of human-computer interaction, they can have a voice in shaping the design of computer tools.

Students also learn about people who apply sophisticated knowledge of CS to advance their work in various fields. Some examples are drawn from our local community. Other, more widely-known, examples include the artist Lillian Schwartz, a pioneer in the use of computers for art analysis and the first computer artist whose work was exhibited at MOMA; psychologists Susan Lederman and Roberta Klatzky who are researching electronic storage and interpretation of information obtained through the sense of touch; and computer scientist Przemyslaw Prusinkiewicz (better known as "Dr. P"), who has developed computer tools for modeling and visualizing plant development.

CSTA Voice ISSN: 1555-2128

CSTA Voice is a publication of the Computer Science Teachers Association.

CSTA Voice is a quarterly publication for members of the Computer Science Teachers Association. It provides analysis and commentary on issues relating to K-12 computer science education, resources for educators, and information for members. The publication supports CSTA's mission to promote the teaching of computer science and other computing disciplines.

Change of Address and Membership Questions: Contact Member Services via email at cstahelp@csta.acm.org, or call 1-800-342-6626 (U.S. & Canada) or +1-212-626-0500 (Global).

Reproduction Rights Information: No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical including photocopy, recording or information storage and retrieval system, without permission in writing from the publisher. Exception: permission to photocopy [individual] items for internal or personal use is hereby granted by CSTA.

Criteria for submitting articles: Potential writers for CSTA should send a brief description of the proposed article, estimated word count, statement of value to members, author's name & brief bio/background info, and suggested title to the editor at cstapubs@csta.acm.org. The final length, due date and title will be negotiated for chosen articles.

Notice to Authors Contributing to CSTA Newsletter: By submitting your article for distribution in this publication, you hereby grant to CSTA the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to allow users to copy and distribute the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and CSTA will make every effort to refer requests for commercial use directly to you.

Here is a brief description of three sample modules from our course.

Art: Working with visual data is an engaging way to learn. Since the earliest days of computing, artists have explored using the computer as a tool for creative expression. In this module, students use simple computer generated images as a means to compare methods of data representation such as bitmap and vector formats. This module also explores the notion of recursion—a powerful but challenging CS concept—in a completely visual way.

Biology: The sequencing of the human genome was accomplished using sophisticated computer algorithms. In this module, students learn how fragment assembly, a key step in genome sequencing, can be modeled in a way that makes it amenable to solution using a computer. Also, a digital view of the information

encoded in DNA shows that CS concepts are not limited to silicon technology.

Language: Why are programming languages different from natural languages? To gain insight on this question, students recreate a famous experiment in natural language processing—a computer program called Eliza. Eliza simulates a psychotherapist, who responds to comments entered by program user. Students play with and extend a simple Eliza program, and experience first-hand the challenges of natural language processing. This illustrates the constraints on programming languages that derive from the need for automatic translation into machine language.

Lecture slides, labs, exercises and hands-on activities, as well as a set of learning goals have been developed for each module. For more information, see course materials at www.ugrad.cs.ubc.ca/~cs101/.

Make it Fun!

CS in Elementary School

Dan Frost

Five years ago, at the urging of my daughter's fourth grade teacher, my wife (a Director of Information Technology) and I (a college-level computer science (CS) educator) took on the challenge of teaching CS to fourth graders and making it fun!

This approach meant minimizing lectures and maximizing hands-on activities. It meant making the class more like art or physical education, and less like math or writing. It meant that students should

The “fun principle” is particularly important for a first exposure to CS

feel lots of success. I wanted to promote the same feelings of universal success and recognition in our CS class as was apparent in the display of the student art work on the classroom walls. The “fun principle” is particularly important for a first exposure to CS, given the not-fun reputation our field sometimes has in high school courses.

Author Marc Prensky labels today's children “digital natives,” and I observed that first hand in the fourth grade students. The school's media center had a sufficient number of computers, the students had studied keyboarding in third grade, and they had some experience with word processing and the Web.

I decided that fun, creative, hands-on programming should be a big part of the class. I wanted a language that gave immediate feedback and supported very

short, interesting programs in under 20 keystrokes. Because neither HTML nor

JavaScript seemed fourth-grade friendly, I decided to create a version of BASIC targeted to grade-school children and delivered as a Java applet. I created it with precisely the features I needed to work with children. Being a Java applet, it could run in the computer browser, eliminating difficult installation issues.

The resulting course was indeed a



**Your First Year of CSTA
Membership is FREE!
Join Today
<http://csta.acm.org>**

Contribute to the CSTA Voice

The editorial board of the *CSTA Voice* is dedicated to ensuring that this publication reflects the interests, needs, and talents of the CSTA membership. Please consider sharing your expertise and love for computer science education by contributing newsletter content.

Potential writers for the *CSTA Voice* should send a brief description of the proposed article, estimated word count, statement of value to members, author's name and brief bio/background info, and suggested title to the editor at: cstapubs@csta.acm.org. The final length, due date and title will be negotiated for chosen articles. Please share your knowledge.

Volunteer today!

The *CSTA Voice* welcomes your comments.

E-MAIL: cstapubs@csta.acm.org

PHONE: 1-608-436-3050

FAX: 1-928-505-5220

Letters to the Editor are limited to 200 words and may be edited for clarification.



ACM founded CSTA as part of its commitment to K–12 computer science education.

The Advocate Blog

Summer Show and Tell

What did you learn this summer that's sure to make you a better teacher this fall?

Read what others have to say and post a comment!

Log onto the

CSTA Advocate Blog

blog.acm.org/csta/

MAKE IT FUN!

continued

great deal of fun—for the fourth graders and for my wife and me, also. In each session, we introduced a new programming command, the students tried it out, and then proudly showed off their accomplishments. Several times we have been told that, for some students, our class is the highlight of the week.

In addition to programming activities, students learned about computer components such as input, output, processing, and storage. Most of the students knew the term “algorithm,” so we examined algorithm strategies such as guessing, computing, and trying each possibility. They explored other CS concepts and terms including branching, repeating,

subroutines, and kilo- and mega- prefixes. We put the CS lessons in context through discussions of computing careers and of people who have contributed to technology developments.

The teachers and principal were excited to see the transfer potential of the CS skills students had learned to the critical-thinking and problem-solving skills taught throughout the curriculum.

Over the past five years we've taught CS to about 400 fourth, fifth, and sixth grade students. More recently I created an applet version of the Logo language which is now our language of choice for fourth graders due to its graphics capabilities. The programming languages and support material are available at www.csed.org.

Reaching Students with Disabilities

AccessComputing

Richard E. Ladner

The Alliance for Access to Computing Careers (*AccessComputing*) is working to increase the representation and success of people with disabilities in computing fields, including computer science (CS). To that end, they are providing a variety of resources to teachers for working with students with disabilities.

There is wide recognition that innovation in computing requires a diverse workforce of qualified systems designers, computer scientists, information professionals, software developers, information systems analysts, technology teachers, computing faculty, and other computing professionals. The inclusion of individuals from under-represented groups, including women, racial/ethnic minorities, and people with disabilities, is needed to provide the innovation and differing perspectives needed for the U.S. to be competitive in the global economy. To meet the need for jobs in the computing field, more women, underrepresented minorities and persons with disabilities need to major in computing fields.

The *AccessComputing* program, centered at the University of Washington with partners from other universities, industry, and computing organizations,

is addressing this need with a number of programs aimed at both students and faculty, including:

- Activities for students with disabilities such as workshops, academies, e-mentoring communities, and internships. Workshops and academies encourage students to pursue computing careers. E-mentoring communities include mentors who can guide students to success. The Alliance can help find internships for students.
- Activities for faculty and departments to increase their capacity to fully include students with disabilities in computing courses and programs. Departments can use an accessibility checklist (www.washington.edu/accesscomputing/equal_access_csd.html) to improve accessibility. Capacity-building institutes help faculty and staff make computing offerings welcoming and accessible.
- A Knowledge Base of FAQs, promising practices, and case studies. The Knowledge Base (www.washington.edu/accesscomputing/kb.html) shares strategies for creating more inclusive courses and programs in which students

with disabilities are encouraged to pursue computing fields.

Teachers can become involved in *AccessComputing* in a number of ways. If you are holding a computer camp or other computing event and would like to include students with disabilities, but are concerned about the added expense, consider applying for a minigrant from the Alliance. If you would like the

computing program at your school to be more accessible examine the accessibility checklist. If you have questions about working with a student with a disability, visit the Knowledge Base. If you have students with disabilities suggest they join the e-mentoring community or participate in other alliance activities. To access any of these or other programs, visit the *AccessComputing* website at www.washington.edu/accesscomputing.

Israel Continues to Improve its CS Education Despite Shared Challenges

Judith Gal-Ezer

As is the case in many countries, the Ministry of Education in Israel is examining and revising its existing computer science (CS) education program to ensure that the program remains up-to-date. Unlike many countries, however, Israel already has a rigorous national curriculum and consistent requirements for CS teachers.

The current Israeli program focuses on lasting CS concepts, rather than on changing technology, while emphasizing algorithmic problem-solving. Two versions of CS are offered; a 5-unit program offers an in-depth study of CS and a 3-unit program provides a more general course of study. Conceptual and experimental topics are interwoven throughout both programs.

In order to be certified to teach CS in Israel, teachers must have at least an undergraduate degree in CS. Israeli teachers also receive pre-service and on going in-service training.

Like teachers in many countries, Israeli teachers have experienced a continuing transition of introductory programming languages, moving from Pascal to C and now to Java. Within the next year, these teachers must complete the shift to Java and its required mastery of the object-ori-

ented programming paradigm as well.

Israeli education researchers report that Israel is experiencing gender equity imbalances similar to those causing concern in the United States and in many other countries. The table below describes the CS secondary enrollment and performance trends in Israel from 2003-2005.

It is worth noting that:

- There has been a decrease in the overall number of students.
- Only 32% of the students are female.
- The average final grade is virtually the same for both males and females.

While Israeli girls comprise only 32% of the students successfully completing CS, girls make up 64% of those passing advanced biology, 63% passing advanced chemistry, and 52% passing advanced mathematics.

In addition to ensuring the very best CS curriculum, educators and researchers in Israel are working to answer the question, "Why don't females choose to take CS as an elective, despite the fact that their achievements are similar to those of their male colleagues?"

Year	Number of Israeli Students			Final Average Grade (0-100)	
	Total	Boys	Girls	Boys	Girls
2003	7841	5343 (68%)	2498 (32%)	89.08	89.54
2004	6796	4608 (68%)	2188 (32%)	88.04	88.48
2005	5658	3884 (69%)	1774 (31%)	88.57	88.82

Meet the Authors

Debbie Carter

CS Educator

Debbie teaches CS at Lancaster Country Day School in Pennsylvania. She is a member of the CSTA Board of Directors and a College Board consultant for AP CS.

Dr. Anne Condon

Department of Computer Science,

University of British Columbia

Anne researches theoretical CS and bioinformatics. She is NSERC/ General Motors Chair for Women in Science and Engineering.

Kate Conley

Editor, Learning & Leading with Technology

Kate is the periodicals director at ISTE with a combined 27 years' experience in education and journalism.

Daniel Frost

University of California, Irvine

Dan is a lecturer in CS at UC Irvine and enjoys teaching undergraduates as well as fourth graders.

Judith Gal-Ezer

Open University of Israel

Judith is a CS professor and chairs the Israeli CS curriculum committee.

Dr. Richard Ladner

Department of Computer Science and

Engineering, University of Washington

Richard is the Principal Investigator of *AccessComputing* and works to promote access to computing for people with disabilities.

Christopher Michaud

Director of Fine Arts,

Nebo Elementary School

Mr. Michaud teaches music in Paulding County, Georgia. He also teaches music and technology in the Nebo after-school program.

Anita Verno

Curriculum Committee Chair, CSTA

Anita is an Assistant Professor and Coordinator of IT at Bergen Community College in New Jersey and an author of *A Model Curriculum for K-12 Computer Science*.

Jacqueline Russell

Product Manager, Microsoft

Jacqueline is responsible for the *Visual Studio Express Editions* development tools for hobbyists and students.

Digi-Know

A Model for CS Curriculum

Anita Verno, CSTA Curriculum Chair

Did you know that CSTA offers teaching and professional resources with the click of a mouse? Beginning with this issue, the Voice will feature a CSTA resource or membership benefit in every edition. We launch with the ACM's Model Curriculum and the related *Objectives and Outlines* documents.

A Model Curriculum for K-12 Computer Science 2nd Edition is a framework for a comprehensive approach to computer science (CS) education, providing an overview of concepts, grouped into four spiraling levels:

- Level I: Foundations of Computer Science – recommended for students in grades K-8.
- Level II: Computer Science in the Modern World – recommended for students in grades 9 or 10.
- Level III: Computer Science as Analysis and Design – recommended for students in upper grades.
- Level IV: Topics in Computer Science – recommended for students in upper grades.

The Model Curriculum and the *Objectives and Outlines* documents have been used by educators around the world to answer the question, "What should I teach to ensure my students have the appropriate foundation for employment and/or continued education?" Several countries, states, districts, and schools are using the Model Curriculum to bring their computing curricula and courses into line with its recommended standards.

The Model Curriculum is a framework, a sketch of what a comprehensive K-12 CS program might include. Detailed objectives for the levels are provided in *Objectives and Outline* documents. These supplemental documents for the Levels II and III courses are available on the CSTA web site. They provide learning objectives, detailed focus points, assessment measures, and sample educational activities for each topic.

To further assist teachers in delivering the objectives of each course, CSTA has created the CSTA Source, a Web repository that contains activities, lesson plans, and assessments. These resources are matched to the topics in each course and can be used for both professional development and in-class activities. We will feature the CSTA Source in a future issue.

Visit csta.acm.org and click on "ACM K-12 CS Model Curriculum" under the "Curriculum" heading to download the Model Curriculum and the *Objectives and Outlines* documents for the Level II Course and the Level III Course. Learn how others have applied these documents to enhance computing education for their students by following the link, "Using the ACM Model Curriculum."

Classroom Tools

Announcing the CSTA Source, A Web Repository

Debbie Carter

We are pleased to announce that the CSTA Source: A Web Repository of K-12 Computer Science Teaching and Learning Materials, is now up and running!

To promote sharing among members of our community, CSTA has developed a searchable database of instructional materials, lesson plans, and other resources that have never before been collected in one place for use by all CS teachers. You may have read about this great new resource in previous issues of the Voice, and though we've had some unforeseeable delays, the

repository is now ready for browsing, downloading, and submitting your own resources.

The repository has been seeded with materials from both JETT and TECS workshops. The resources have been grouped into four curriculum levels, with multiple topics within each one, based on the *ACM Model Curriculum for K-12 Computer Science*, and the *AP CS Course Description*. There is also a collection called "Strategies for Implementation," with ideas that span curriculum levels: clubs, equity considerations, and promoting CS. CSTA volunteers have classified and entered these materials into the repository, and there are now over 100 resources of various types: multimedia presentations, lesson plans, assignments (both lab and written), papers, and source code.

These resources are available for download by all CSTA members. To access the repository, visit the CSTA Web site (csta.acm.org), and click on the "CSTA Source" link. You may browse by curriculum classification or search by title, author, keyword, or publication date. You may download any resources that interest you. (You'll need your CSTA member number to download resources.)

But this is just the beginning; the repository will be most useful when many teachers have contributed their own resources. Submitters fill in a form with information about each resource they wish to submit, and then they upload the resource file/files to the repository. They retain the copyright on their materials, while giving CSTA the rights to allow distribution to other members.

The CSTA Web Repository page provides guidelines for classifying and submitting your resources, as well as an optional template for organizing the information before you enter it online. All submissions will be reviewed by a team of expert educators before appearing in the repository.

Which will you do first: look in the repository for resources or prepare some of your own for submission? We are eager for you to do both.

Spotlight

Starting with the Beginner Developer Learning Center

Jacqueline Russell

In an effort to address the increasing digital divide and the downturn of interest in computer science, Microsoft is attempting to revitalize programming with the release of free tools and new learning materials.

According to Microsoft's Jacqueline Russell, the software development landscape in 2005 was typified by:

- a) The continuing steady decline in university computer science (CS) enrollment.
- b) The information technology (IT) outsourcing trend plus the developer talent importing trend.
- c) Rapid advances in technology and software plus an equally rapid decline in the number of people who actually knew how technology worked!

Microsoft saw these things and decided to do something about it—provide free developer tools and learning resources. The purpose of releasing these products was to help reinvigorate the passion for programming that was so evident back in the days of BASIC when anyone and everyone could pick up a magazine and learn to create a little desktop utility. Unfortunately, since that time there has been a growing divide between "users" and "professional developers."

In 2005 Microsoft released a free version of programming tools called the Visual Studio Express Editions. Since then, there have been over 13 million downloads of these tools, and a large enthu-

SHOW ME THE NUMBERS An AP Snapshot

Number of students who wrote the A and AB CS Exam in 2006 - 19,601

EXAMINEES BY GRADE LEVEL:

9th grade: 1.4%
10th grade: 14.5%
11th grade: 36.6%
12th grade: 42.4%
Other: 5.2%

EXAMINEES BY GENDER

Male: 84%
Female: 16%

SOURCE: www.collegeboard.com/prod_downloads/about/news_info/ap/2007/2007_ap-report-nation.pdf

community has sprung up around recreational development, learning and sharing through the Coding4Fun site and others.

Last March, Microsoft also launched the Beginner Developer Learning Center (BDLC) website which contains free lessons and tutorials for anyone who is just learning to program. All lessons in the Beginner Developer Learning Center (msdn.microsoft.com/vstudio/express/beginner/) are organized into two development tracks: Windows Development and Web Development. In addition, they are further categorized into tiers of difficulty to make course progression and selection more straightforward.

Russell believes that the second digital divide of our age is one of knowledge, not access: “Resources such as the BDLC will help to bridge the gap between the creators and the consumers of software applications and services in order to encourage everyone to participate in this digital age”. She encourages educators to try the resources and to send any feedback/comments/suggestions to her attention at jacqueline.russell@microsoft.com

Our Story

CS for Elementary Students

Christopher Michaud

In my role as a music teacher at Nebo Elementary School in Paulding County, Georgia, I use music and technology to teach students to not simply consume media, but to create and share media with the world. It all takes place in an after-school technology club where fourth and fifth grade students develop computing skills by creating digital media.

I wanted to move my students beyond using office applications and the Internet to a deeper understanding of, and fluency with, computer science (CS) and programming concepts. Scratch from MIT provided a starting point for teaching the basics of computer science through story-telling using interactive gaming.

Scratch is a free, block-based programming environment created by MIT to produce media and teach programming concepts. Using drag-and-drop techniques to join together block icons representing code, students create stories in which programmable objects called “sprites” move and interact. Scratch’s graphic block-based environment provides an excellent platform for introducing CS concepts such as objects, methods, and conditional statements.

In the first Scratch lesson, students created a sprite object and assembled the code to move the sprite about the screen in just 15 minutes! In the first two hours, students created multiple sprites that used random numbers to move on their own and basic conditional statements for interaction and scoring. They also experimented with the Scratch sound and painting functions. We posted our early games to the Scratch online gallery after only

two weeks of learning.

After this initial success, I challenged my students with a larger team project. The assignment was to create an interactive project in which students must:

- Create a story with characters, settings, conflict and resolution.
- Design the solution as a video game.
- Create the sprite characters.
- Create the game program code with Scratch.
- Create a webpage advertisement for the game.

Over the course of five weeks, I introduced programming concepts and strategies such as creating and modifying a sprite, making a sprite move on its own with random numbers and “following” strategies, keeping score, and sensing with if-then statements.

Soon, the students came up with the solutions to their own programming problems. They discovered how to have characters follow one another and to grab and carry other objects, to score accurately, to change difficulty levels, to shoot projectiles, and to add sound effects. Their learning linked these game strategies to math concepts such as positive and negative numbers, grid coordinates, directionality, random numbers, and variables.

My students presented their Scratch projects at the National Educational Computing Conference (NECC 2007) in Atlanta showcasing what 10 and 11 year olds can create through computer programming. They demonstrated Scratch to create objects and write procedures. The Nebo students’ knowledge and energy inspired other teachers, students, and administrators to learn more about Scratch and to use computer programming as a tool to teach the Georgia Performance Standards.

I believe that students should leave the fifth grade having had a basic introduction to CS through a visual programming language such as Scratch. Learning CS concepts teaches students logical thinking and higher-order thinking skills. A visual programming language encourages kinesthetic thought in a system where students can examine and edit their creations as they change over time. CS activities also provide an opportunity for project-based and integrated learning in accordance with the Georgia Performance Standards and others. Finally, students with an understanding of the structure and flow of computer programs develop a deeper fluency with software and hardware. This fluency enables students to express their creativity and share their creations with the world.

Nebo students’ Scratch projects can be accessed at the “Nebomusic” Gallery at scratch.mit.edu/galleries/99. More information about the Nebo Department of Fine Arts can be found at www.nebomusic.net.

Bits and Bytes

SIGCSE Opportunities

SIGCSE 2008 will be held March 12-15, 2008 at the Oregon Convention Center, located along the Willamette River, in beautiful Portland, Oregon. Information about the symposium deadlines and the SIGCSE 2008 conference is available at www.cs.duke.edu/sigcse08/.

Opportunities for becoming a reviewer for the ACM SIGCSE conference can be accessed at www.cs.grinnell.edu/~sigcse/sigcse2008/reviewerRegistration.shtml.

We Goofed!

The URL in the SeeING Gender article in the June, 2007 issue of the CSTA Voice was incorrect. The correct URL is www.meac.org/Resources/ed_services/SG_WEB/SeeingGender/



MARK YOUR CALENDAR

International Computing Education Research Workshop (ICER)

September 15-16, 2007 in Atlanta, Georgia
www.cc.gatech.edu/conferences/icer2007/

Consortium for Computing Sciences in Colleges (CCSC-Midwest)

September 28-29, 2007 in Hamilton, Ohio
www.ccsc.org/midwest/conference

Frontiers in Education Conference 2007

October 10-13, 2007 in Milwaukee, Wisconsin
fie.engrng.pitt.edu/fie2007/

Consortium for Computing Sciences in Colleges (CCSC-Eastern)

October 12-13, 2007 in Patchogue, New York
www.sjcnny.edu/ccsce2007

CSTA Sessions:

Supporting K-12 Computer Science Education
(Anita Verno)

Workshop: Introduction to Alice
(Stephen Cooper)

Consortium for Computing Sciences in Colleges (CCSC-Northwest)

October 12-13, 2007 in McMinnville, Oregon
www.ccsc.org/northwest/2007/

Tapia Celebration of Diversity in Computing Conference

October 14-17, 2007 in Orlando, Florida
www.richardtapia.org/2007/

Grace Hopper Celebration of Women in Computing

October 17-20, 2007 in Orlando, Florida
gracehopper.org/2007/

Consortium for Computing Sciences in Colleges (CCSC-Rocky Mountain)

October 19-20, 2007 in Orem, Utah
www.ccsc.org/rockymt/

Consortium for Computing Sciences in Colleges (CCSC-Southeast)

November 9-10, 2007 in Myrtle Beach, South Carolina
cs.furman.edu/ccscse/conference.php?year=21st

Informatics Education in Europe (IEEII)

November 29-30, 2007 in Thessaloniki, Greece
www.seerc.org/ieei2007/

SIGCSE 2008

March 12-15, 2008 in Portland, Oregon
www.cs.duke.edu/sigcse08/
Submission Deadline: September 7, 2007

National Educational Computing Conference 2008 (NECC)

June 29-July 2, 2008 in San Antonio, Texas
web.uoregon.edu/ISTE/NECC2007/program/NECC2008.php
Submission Deadline: October 3, 2007

For the current JETT workshop schedule visit jett.acm.org/

For the current TECS workshop schedule visit tecs.acm.org/

The complete ACM Calendar of Events is available at campus.acm.org/calendar/

Keep up-to-date with the Consortium for Computing Sciences in Colleges (CCSC) at www.ccsc.org/

RESOURCES

Here's more information on topics covered in this issue of the *CSTA Voice*.

Page 1: CS & IT Symposium www.csitsymposium.org

Page 2: Interdisciplinary CS 101 www.ugrad.cs.ubc.ca/~cs101/

Page 3: Computer Science Education Association www.csed.org

Page 3: Marc Prensky www.marcprensky.com/

Page 4: AccessComputing www.washington.edu/accesscomputing

Page 6: CSTA Source csta.acm.org/Resources/sub/WebRepository.html

Page 6: ACM Model Curriculum csta.acm.org/Curriculum/sub/ACMK12CSModel.html

Page 6: Beginner Developer Learning Center msdn.microsoft.com/vstudio/express/beginner/

Page 6: How-To Reference Library msdn.microsoft.com/vstudio/express/beginner/reference/

Page 6: Kid's Corner msdn.microsoft.com/vstudio/express/beginner/kids/

Page 7: Scratch www.scratch.mit.edu

Page 7: Nebomusic Gallery scratch.mit.edu/galleries/99

Page 7: Nebo Elementary Department of Fine Arts www.nebomusic.net

Page 7: SIGCSE www.cs.duke.edu/sigcse08/



Computer Science Teachers Association

The Advocate Blog

Talk to us!
blog.acm.org/csta