Variations on a Theme: 
Role of Media in Motivating Computing Education

Mark Guzdial (Moderator)  
School of Interactive Computing, 
Georgia Tech 
85 5th Street NW 
Atlanta, GA 30332-0760 
+1-404-894-5618 
guzdial@cc.gatech.edu

David Ranum, Brad Miller  
Luther College 
Computer Science 
700 College Drive 
Decorah, IA 52101 
+1-901-GONORSE 
{ranum,bmiller}@luther.edu

Beth Simon  
University of California, San Diego 
Computer Science & Engineering Dept. 
La Jolla, California 92039-0404 
+1-858-534-5419 
bsimon@cs.ucsd.edu

Samuel A. Rebelsky, Janet Davis  
Grinnell College 
Department of Computer Science 
Grinnell, IA 50112 
+1-641-4410 
{rebelsky, davisjan}@grinnell.edu

Barbara Ericson  
College of Computing, Georgia Tech 
801 Atlanta Drive 
Atlanta, GA 30332 
+1-404-385-2108 
ericson@cc.gatech.edu

Deepak Kumar, Doug Blank  
Bryn Mawr College 
Department of Computer Science 
Bryn Mawr, PA 19010 
+1-610-526-7485 
{dkumar, dblank}@brynmawr.edu

Categories and Subject Descriptors  
K.3.2 [Computing Milieux]: Computers and Education – Computer and Information Science Education

General Terms  
Design, Experimentation

Keywords  
Multimedia, Education, Motivation, Media Computation

SUMMARY

The SIGCSE community has been exploring the role of multimedia to enhance computing education since the earliest algorithm visualization systems and studies [1]. We have been exploring for years how images, sounds, and video might improve how students learn computer science. Media Computation is a shift in focus [2]. Where algorithm visualization presents information to the student to facilitate their understanding, media computation is about having students manipulate media as the data for their programming, i.e., as the focus of the course activities. Students in media computation produce new images, sounds, and video. The reason for the media is to increase student engagement and motivation. We aim to show that computer science is about more than numbers and strings. Computer science is also about creative expression.

The original media computation work focused on using media to motivate non-computing majors [3]. The role of media in motivating student learning for computing education has broadened. Inventive teachers are using media computation for lots of different kinds of students, at different kinds of institutions, with a range of languages and toolkits.

This special session is a mixture of “Five Minute Madness,” science fair, and art gallery. Each participant will present how he or she is using media to motivate student learning, and some student work will be available for audience inspection. To be specific, each participant (or participant group) will have five minutes (with an iron hand on the stopwatch):

- To describe why the participant is using media in his or her class. What's the advantage? What does it buy you and your students? Do you plan to keep using media as an introductory context?
- To describe in detail one assignment that your students really work on, at your institutional setting.
- To show two different student artifacts, with at least one being an output from that assignment.

The rest of the time will be left for Q&A with the audience. Participants are asked to provide a poster with an example of an assignment definition and some student work, to be displayed in the room. Participants may reference their poster in their session or in response to questions. Audience members are invited to peruse the student art gallery before and after the session.

1. PARTICIPANTS

David Ranum and Brad Miller, Luther College.

In the Introduction to Computer Science I course at Luther College, we use image processing as an example of basic object-oriented programming. Digital images are treated as objects made up of a grid of RGB pixels, each of which is also an object. Students use methods such as getWidth, getHeight, getPixel,
setPixel, getRed, and setRed to manipulate the images. Two interesting projects are image enlargement and edge detection. Enlarging an image requires students to find mapping patterns to copy pixel values from the smaller image to the larger. This also leads to a need for image smoothing to remove the “blockiness” that occurs. Edge detection requires students to compute gradients that surround a pixel. To do this we introduce the concept of a weighting mask and a convolution pattern. Pixels found to exist on a large gradient are considered to be edge pixels and are colored accordingly.

Beth Simon, University of California—San Diego.

UCSD is a large research-intensive university which adopted media computation in Java during the 2008-2009 academic year. The traditional introductory 10-week course for students with “no prior programming experience” which targets approximately 2/3 of the incoming majors used this approach. The goal of the change was to address a large attrition rate in majors -- by the beginning of the second year 32% of incoming majors are no longer taking computing courses. Additionally, since the major was only recently re-opened to allow students to change into the computing major (having been locked down to incoming freshman only during the dot.com boom), we sought an approach that was likely to appeal to students “giving computer science a try” – while still providing them an authentic programming experience and keeping them on track to continue in major sequence courses. Response so far has been overwhelmingly positive.

Barbara Ericson, Georgia Institute of Technology (and Lakeside High School).

I am co-teaching an Advanced Placement Computer Science A high school course using Alice and Media Computation. We will be using the new book “Exploring Wonderland: Java Programming with Alice and Media Computation” [2]. We will be using Alice to introduce new computing concepts and then facilitating the transfer of those concepts to Java using Media Computation. You can think of Media Computation as the special effects studio for Alice movies. It is how we will create sound clips so that we don't violate copyright by using a complete song in an Alice movie. We can also merge live action and Alice movies using chroma-key.

Sam Rebelsky and Janet Davis, Grinnell College.

In the MediaScripting approach to media computation that we use at Grinnell, students write Scheme scripts to create and manipulate images within the GIMP, an open-source raster graphics application. Because they script within the GIMP, students are able to use a mixed-mode approach to image creation and manipulation: Sometimes they write scripts, sometimes they do things “by hand”, with the tools of the application. This mixed-mode approach helps motivate programming problems; often, students will find that they want to write scripts to replace what they do by hand - to ensure precision, to encode a sequence of actions so that they can do it again, and even to experiment. We believe that this approach encourages students to think about how the ability to program can help them in a variety of contexts. Scheme allows us to have students think about images in multiple ways, for example, not only do they use an imperative approach to scripting the tools, they also model images as functions from positions to colors.

Deepak Kumar and Doug Blank, Bryn-Mawr College.

In the introductory computing course developed by IPRE (Institute for Personal Robots in Education), students learn computing by exploring the context of programming robot behaviors [4]. Each student's personal robot includes a camera, a gamepad controller, speech and sound/music synthesis. This allows for a rich set of experiences for students. They learn to capture, and manipulate camera images for many applications including robot navigation, object tracking, as well as creating creative image transformations. Several aspects of media manipulation find a natural place in this context. The Python-based Myro toolkit used in this course was designed to cater to such pedagogical goals. In another introductory course we are experimenting with media and creative computing aspects in using the Processing programming language.

2. REFERENCES


