Poster Session 2

Combining Flipped Learning with Gamification to Improve Student Performance in a Data Structures Course

Darina Dicheva, *Winston-Salem State University*
Contact: dichevad@wssu.edu

Data Structures is one of the fundamental computing courses but also among those with the highest rate of drops and failures. Its importance for a timely completion of Computer Science college programs implies the need of employing innovative instructional methods and technologies to improve student engagement and performance. This poster presents an instructional method that leverages strategies from both flipped learning and gamification. The flipped classroom has been shown to effectively support active learning activities. Moreover, flipping a Data Structures classroom allows introducing programming labs in class, which is not typical for the standard way of teaching the course. However, the success of flipped learning depends crucially on the proper engagement of students in and outside of class. Here gamification can help. Gamification, defined as the use of game design elements in non-game contexts, has shown promises in shaping user behavior in desirable directions, which can be used to increase students' motivation and engagement in the learning process. The poster describes a gamified flipped learning approach and the author's experience and findings in using it in a Data Structures course at a minority university. In the proposed method, the in-class work features group problem solving and pair-programming labs. The employed gamification elements include accruing points, virtual currency, rapid feedback, unlocking content, freedom to fail, and social engagement. The measures used in assessing the effectiveness of the proposed approach include students' grades, information about class attendance and submission of lab assignments, and an attitudinal survey.

Keywords: Flipped learning; Gamification**;** Data Structures

DOI: http://dx.doi.org/10.1145/2839509.2850529

Assessing the Effectiveness of Experiential-Learning-Based Teaching Tools in Cybersecurity Courses

Xiaohong Yuan, *North Carolina A&T State University*
Jinsheng Xu, *North Carolina A&T State University*
Huiming Yu, *North Carolina A&T State University*
Junghee Kim*, North Carolina A&T State University*
Taehee Kim, *North Carolina A&T State University*
Contact: xhyuan@ncat.edu

The poster describes our project of Assessing the Effectiveness of Experiential-Learning-Based Teaching Tools in Cybersecurity Courses. We are assessing the effectiveness of experiential-learning-based teaching tools for 10 cybersecurity topics in five cybersecurity courses. For each topic, two teaching methods are selected: the control group teaching method, and the experimental group teaching method. The two teaching methods are compared to answer one of the two research questions: (1) Is using an experiential-learning-based teaching tool more effective in improving student learning than the traditional teaching method without using the teaching tool? (2) Is one experiential-learning-based teaching tool more effective than another experiential-learning-based teaching tool? We will assess the effectiveness of the teaching methods via three measures: (1) improvement in student learning outcomes, (2) improvement in student motivation in learning the topic, and (3) improvement in the student experience such as student enjoyment, satisfaction, and perceived difficulty in learning the topic. The knowledge gained from this research can be used by cybersecurity educators at other institutions to use effective teaching tools to improve cybersecurity education practices, which has the potential to increase the number of students well-prepared for entering cybersecurity careers.

Keywords: experiential-learning-based teaching tools; cybersecurity; assessment

DOI: http://dx.doi.org/10.1145/2839509.2850550

Teaching and Learning in an Introductory Undergraduate Programming Class: A Reflective Autoethnography

S. Zahra Atiq, *Purdue University*
Contact: satiq@purdue.edu

In scholarship of teaching and learning studies (SoTL), computer science (CS) educators have investigated the effectiveness of their teaching methods on students, but they have rarely investigated their own professional development. Some educators do examine their own practices through a process of reflection, but these informal self-reflection efforts have lacked scholarly rigor. A more rigorous study could be provided through the use of autoethnography. This poster presents an autoethnographic study by the author on the teaching of introductory computer programming courses. Through this study, which used five years of data, the author arrived at three main conclusions. First, the author found that students were able to learn new programming languages because the author emphasized self-learning and life-long learning in the classroom. Second, the author found that peer learning, in which one student explains ideas to another, benefits both students. Third, the author found that when educators mentor, support, and guide students, the students' motivation, engagement, and retention increase. Although these findings about effective practices confirm the results of previous studies, the autoethnographic method provides a new, credible way for educators to reflect and inform their practices.

Keywords: autoethnography; computer programming; reflection; research to practice

DOI: http://dx.doi.org/10.1145/2839509.2850553

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.

Eighteen Hours of Code with Fifth Grade Students

Katie Davis, *California Polytechnic State University- San Luis Obispo*
Zoë Wood, *California Polytechnic State University- San Luis Obispo*
John Wilcox, *Peabody Charter School*
Contact: kdavis22@calpoly.edu

Integrating programming and computer science into the K-12 curriculum is of national importance. This poster reports on our experience creating an eighteen-week lab curriculum for fifth grade students. Using Processing, an open source programming language and IDE that is built on the Java programming language, we developed and taught eighteen separate lab components introducing basic coding concepts appropriate for fifth grade. These lab modules were developed by a team consisting of a fifth grade teacher, a computer science master's student, and a computer science professor. The labs were taught to over 100 fifth grade students in the 2014-15 academic school year at a public school, Peabody Charter School. All coding labs produced visual output, either static scenes or animations. Topics included basic syntax with a large focus on two dimensional spatial reasoning, but also included three different modules focused on concepts from the fifth grade science curriculum. All curriculum is available at http://users.csc.calpoly.edu/~zwood/Outreach/PCS/PCS5.html. This poster breaks down the basic lab components developed and presents challenges and positive outcomes. In general, students enjoyed the exercises; the teacher observed that all students were eager to complete the labs. Prior to this experience the teacher had no programming knowledge and learned alongside the students each week. He continues to teach the Processing modules in the fifth grade for the 2015-16 school year.

Keywords: introductory programming; computer science education; K-12 education

DOI: http://dx.doi.org/10.1145/2839509.2850554

Multifaceted Efforts to Create an Inclusive Environment and Increase Diversity

Perry Fizzano, *Western Washington University*
David Hartenstine, *Western Washington University*
Contact: perry.fizzano@wwu.edu

The goal of increasing the representation of women and minorities in Computer Science is widespread and one to which we are strongly committed. There are several notable academic programs and professional organizations that are making headway on this goal themselves and are making resources available to others. While we have learned from others, we have also developed some novel approaches appropriate to our institution we feel are suitable for others.

In this poster, we outline the best practices that we've adopted and describe our novel approaches. Some of these efforts were funded by an NSF S-STEM grant which provided scholarships and supporting services for females pursuing degrees in Computer Science or Mathematics. Other ideas spawned independently of that grant and require few resources. Our efforts include: (1) actively recruiting female students (2) early advising and peer mentoring (3) creating a supportive department culture, paying special attention to underrepresented groups and first-generation students (4) developing teaching practices to increase freshmen interest and retain students through graduation (5) encouraging students to become members of the larger community (6) promoting professional and leadership development. We describe how the novel approaches we've developed contribute to these recognized best practices, report on our actions, and include our qualitative assessment of improved climate and quantitative results related to improved recruitment and retention. The human and financial resources required for each action item are indicated. Our ultimate goals are to share our approaches, learn from our colleagues, and to foster continued engagement.

Keywords: Diversity, Women in CS

DOI: http://dx.doi.org/10.1145/2839509.2850555

Deepening Learning in High School Computer Science through Practices in the NGSS

Marie Bienkowski, *SRI International*
Contact: marie.bienkowski@sri.com

High school computer science teaching can be informed by how science and engineering practices are defined in K-12 curriculum standards. In the United States, the Next Generation Science Standards (NGSS) are seen as broadly stated and widely accepted statements of what children should know about science and engineering to be literate citizens and to prepare for careers in STEM. While the NGSS do not identify practices for computer science, our experience is that the broadly defined engineering practices can serve as an appropriate framework for thinking about computational work. Our interpretation differs from efforts that use computation to teach science or mathematics, instead we apply NGSS approaches to teach computer science. We anticipate that K-12 teachers who are working with these standards as well as with the Common Core State Standards (CCSS) will find, as we have, a crossover from the emphasis in these standards on inquiry, argumentation, and overall deeper learning to the pedagogical orientation of curricula such as the high school introductory course Exploring Computer Science (ECS). This poster will show how a portion of the NGSS, carried over from the precursor Science and Engineering Framework and called the "Condensed Practices" can be used with activities in ECS to deepen student learning and help teacher adoption. Handouts will be available showing (1) knowledge, skills, and abilities for computational thinking, (2) curriculum guidance for teachers wishing to use the NGSS to inform ECS teaching and (3) suggestions for professional development in NGSS for computational thinking for all K-12 teachers.

Keywords: Exploring Computer Science; High School Computer Science; Engineering Design Practices

DOI: http://dx.doi.org/10.1145/2839509.2850557

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.

Replicating a Validated CS1 Assessment

Miranda Parker, *Georgia Institute of Technology*
Mark Guzdial, *Georgia Institute of Technology*
Contact: miranda.parker@gatech.edu

Validated assessments are important for teachers and researchers. A validated assessment is carefully developed to make sure that it is measuring the right things. Computing education needs more and better validated assessments. Validated assessments provide instructors with insight on how their students are doing in their class and provide researchers with insight on whether certain technologies and interventions are successful. Building high-quality, validated assessments is difficult. However, it is possible to *replicate* an existing validated assessment, and the new assessment can be validated against the original assessment. We need mechanisms to replicate assessments so that we can build more and more varied assessments for different audiences. We developed the Secondary CS1 Assessment (SCS1) as an isomorphic version of a previously validated assessment instrument for introductory computer science. In this poster we provide an overview for the process of replicating an existing valid knowledge assessment and validating the replication. Handouts will be provided with information about how to access and use the SCS1 Assessment.

Keywords: Assessment; CS1; validity; replication

DOI: http://dx.doi.org/10.1145/2839509.2850559

Assessing the Development of Computer Science Pedagogical Content Knowledge in the TEALS Program

Yvonne Kao, *WestEd*
Leigh Ann DeLyser, *CSNYC*
Aleata Hubbard, *WestEd*
Contact: ykao@wested.org

One of the critical barriers to increasing pre-collegiate computer science course offerings in the U.S. is a lack of qualified computer science teachers. Programs such as TEALS, a teacher preparation program pairing high school teachers with computing professionals to offer CS courses, provide opportunities for in-service teachers to gain experience teaching computer science. However, it is not clear whether the high school teachers develop sufficient pedagogical expertise to sustain high-quality computer science course offerings at their schools. Furthermore, the field of computer science education lacks valid and reliable ways of measuring pedagogical content knowledge (PCK), a construct that describes the knowledge teachers need for effective instruction. In this poster, the authors present these results from the first year of a three-year NSF grant to study how TEALS participation influences novice computer science teachers' PCK: 1) a theoretical framework describing the critical components of CS PCK, 2) the results of the first field test of a CS PCK assessment, including the psychometric properties of the assessment, and 3) a comparison of how teachers performed on the assessment at the beginning and end of their first year of computer science teaching and how they performed relative to their computing professional mentors.

Keywords: computer science education; pedagogy; pedagogical content knowledge; high school; teacher preparation

DOI: http://dx.doi.org/10.1145/2839509.2850560

A Curiosity-Driven System for Developing Coding Literacy

Neeraj Chatlani, *Rollins College*
Daniel S. Myers, *Rollins College*
Contact: dmyers@rollins.edu

Coding literacy is the ability to understand a written computer program and interpret its functionality and output. Literacy is a valuable skill for programmers at all levels, because understanding written code requires developing and applying mental models of program execution. Previous work has shown that explicit instruction in program literacy is beneficial for new computer science students and aids the development of algorithmic thinking. This poster summarizes the authors' work-in-progress developing COLT: the Coding Literacy Trainer, a web-based adaptive tutorial system that provides instruction in the fundamentals of coding literacy and program interpretation to new computer science students. In addition to its pedagogical applications, COLT serves as a development platform for a novel theoretical foundation for adaptive teaching systems based on the concept of intrinsic curiosity. Inspired by the work of Lee et al. in the field of developmental robotics, a curiosity-driven system explores its complete knowledge environment in way that continually maximizes its learning progress. Thus, learners are driven to explore areas where they are currently making the greatest advances, while avoiding regions of the knowledge space that are either too simple to be interesting or too challenging to be approachable at the current time. The poster summarizes the theoretical background and implementation of the COLT system in a clear, easy-to-read format. A web-based version of COLT is currently under active development and slated for an open-source release in the spring of 2016.

Keywords: Coding literacy; Adaptive learning systems; Intrinsic curiosity; Computer science education

DOI: http://dx.doi.org/10.1145/2839509.2850561

Teaching Software Engineering Skills in CS1.5: Incorporating Real-world Practices and Tools

Sarah Heckman, *NC State University*
Jason King, *NC State University*
Contact: {sarah_heckman, jtking}@ncsu.edu

Students learn best in environments where they can meaningfully engage with materials that emulate real-world scenarios. Incorporating software engineering best practices and supporting tools in introductory courses provides students the opportunity to engage in course materials as a novice member of the profession. We support student engagement with industry tools to support software engineering best practices for tutorials, in-class labs, and

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.

programming projects. *The goal of the research is to improve student learning, engagement in the course and profession, and retention through the use of software engineering practices and tools that introduce students to the software engineering profession.* A prior study on the incorporation of in-class laboratories, supported with software engineering best practices, on linear data structures showed an increase in engagement, but did not show a difference on student learning when compared with active learning lectures. We are currently expanding the study by incorporating in-class laboratories across a full semester of a CS1.5 class at NC State University. The poster presents the preliminary results from Fall 2015.

Keywords: software engineering best practices; CS1.5; situated learning

DOI: http://dx.doi.org/10.1145/2839509.2850562

Designing and Refining of Questions to Assess Students' Ability to Mentally Simulate Programs and Predict Program Behavior

Ashish Aggarwal, *University of Florida*
Christina Gardner-McCune, *University of Florida*
David S. Touretzky, *Carnegie Mellon University*
Contact: ashishjuit@ufl.edu

Mental simulation is an important skill for program understanding and prediction of program behavior. Assessing students' ability to mentally simulate program execution can be challenging in graphical programming environments and on paper-based assessments. This poster presents the iterative design and refinement process for assessing students' ability to mentally simulate and predict code behavior using a novel introductory computational thinking curriculum for Microsoft's Kodu Game Lab. We present an analysis of question prompts and student responses from data collected from three rising 3rd - 6th graders where the curriculum was implemented. Analysis of student responses suggest that this type of question can be used to identify misconceptions and misinterpretation of instructions. Finally, we present recommendations for question prompt design to foster better student simulation of program execution.

Keywords: Computational Thinking; Kodu Game Lab; Program Simulation; Mental Simulation; K-12 CS Education & Curriculum; Instruction Design

DOI: http://dx.doi.org/10.1145/2839509.2850563

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.

Megas and Gigas Educate (MaGE): A Curricular Peer Mentoring Program

Heather Pon-Barry, *Mount Holyoke College*
Audrey St. John, *Mount Holyoke College*
Becky Wai-Ling Packard, *Mount Holyoke College*
Barbara Rotundo, *Mount Holyoke College*
Contact: ponbarry@mtholyoke.edu

The Megas and Gigas Educate (MaGE) program is a peer mentoring program being developed at Mount Holyoke College, a liberal arts college for women, for the introductory CS curriculum. Consistent with national trends, interest in CS is rising rapidly; current resources cannot meet demand while maintaining quality feedback and pedagogy. Supported by a Google Computer Science Capacity Award, MaGE has three main objectives: (1) to triple enrollment capacity over 3 years in introductory courses; (2) to increase enrollment and retention for women and other underrepresented groups; (3) to train students to educate, mentor, and support others in inclusive ways. Trained undergraduate students act as peer mentors to beginner students, providing close interaction and assisting with feedback. MaGE is currently being piloted in the introductory CS1 course. Enrolled students bring varying interests, including Art, History, Biochemistry, Economics and Engineering; most students have no prior programming experience. Each CS1 student is assigned a peer *Giga Education Mentor* (or GEM) in a 9:1 ratio. GEMs have undergone a rigorous training course that raises awareness of the role of social identity in learning, emphasizes active learning within computer science, and provides preparation for being technical peer mentors. While research supports the need for culturally-sensitive, inclusive training as part of the curriculum, we know of few peer-based models in CS that explicitly include this education. By building a diverse set of peer role models and connecting with the pre-existing co-curricular *Megas and Gigas* mentoring program, MaGE seeks to effectively engage underrepresented students in computing.

Keywords: Peer mentoring; diversity and inclusion; innovative curriculum design; recruiting and retaining students

DOI: http://dx.doi.org/10.1145/2839509.2850564

Pixels, Post-It's® and CS Principles

Jeffrey L. Popyack, *Drexel University*
William M. Mongan, *Drexel University*
Contact: popyack@drexel.edu

We describe an "unplugged" activity that exposes data structures and algorithms involved in image representation and compression, using multi-colored Post-It Notes®. We use Post-It's to illustrate how color images can be encoded and decoded so they can be stored efficiently on a computer and/or transmitted efficiently across a network. This provides an intuitive illustration of bitmaps and the GIF (Graphics Interchange Format), optimized with Lempel-Ziv compression. The algorithms are simple to explain and comprehend, and simultaneously appeal to the participants' sense of art and creativity while demystifying advanced computing concepts and computer graphics.

We describe two sample exercises which make use of these concepts. In one exercise, teams of participants create messages with Post-It's, encode them and send them to other teams for decoding. In another exercise, a larger image is encoded and broken into letter paper-sized smaller images to be decoded and reassembled by a group.

This activity has proven to engage participants with a wide variety of backgrounds, and provides the background needed for further exploration and discussion of Computer Science Principles, specifically the Big Ideas of Creativity, Abstraction, Data and Information, and Algorithms.

Keywords: CS Principles; Unplugged; Hands-On

DOI: http://dx.doi.org/10.1145/2839509.2850565

Adventures in K-5 STEM Outreach Using the NAO Robot

Steve Hadfield, *United States Air Force Academy*
Chris Coulston, *Pennsylvania State Univ. - Erie*
Marissa Hadfield, *Academy School District 20*
Lillian Warner, *United States Air Force Academy*
Contact: steven.hadfield@usafa.edu

The humanoid NAO robot continues to win both hearts and imaginations with its lifelike appearance and behaviors. Its consistent growth in popularity, an increasing wealth of free behaviors, and the intuitive Choregraphe development environment provide educators and developers with exceptional opportunities to motivate interest in STEM disciplines and breach impediments such as perceptions that such technology is simply too hard. In this poster, the authors discuss a variety of experimental uses of the NAO robots for K-5 STEM Outreach. Initial development and demonstrations focused on generating enthusiasm for both robotics and programming. Dancing and exercising behaviors from Notre Dame University's F.U.N. Lab and Aldebaran Robots easily integrated into voice-controlled Choregraphe demonstration scripts. Faculty and undergraduate students directed this enthusiasm to motivate engagement in Hour of Code programming activities. The team also utilized the NAO robots in small group settings within a K-5 Response To Intervention (RTI) program where demonstrations were followed by having the children interactively experiment with the robots stimulating imagination, creativity, curiosity, and problem solving skills as well as confidence and self-esteem. Next the RTI children actually programmed the robots using a story-based methodology and the powerful while intuitive building block programming constructs of Choregraphe. The team's on-going development efforts focus on expanding the repertoire of available behaviors to include interactive math games and foreign language educational dialogs. Results from use of these new behaviors will be presented at the SIGCSE Symposium. The team is also working to measure attitudinal, conceptual understanding, and math and language skills improvements.

Keywords: STEM; Outreach; K-5; robots; RTI; NAO

DOI: http://dx.doi.org/10.1145/2839509.2850566

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.

Engaging School Counselors, Creating Computing Allies

Sarah Hug, *University of Colorado Boulder*
Jane Krauss, *NCWIT*
Contact: jkrauss@mahonia.us

When counselors become advocates for computing they can have significant impact in recruiting youth in the field through their administration of course assignments and engagement in career exploration with students. A recent study of counselors' academic backgrounds, work priorities, and current perspectives and practices regarding advising in computing has implications for educators and others interested in building cooperative relationships with counselors. The study took place as an activity of the *Counselors for Computing (C4C)* program. C4C leverages the National Center for Women & Information Technology (NCWIT)'s organizational membership, applies evidence-based approaches for professional development, and monitors and adjusts practices to make inroads into CS education through school counselor engagement. In an effort to tailor existing professional development for counselors to the realities of their complex roles in schools, C4C asked 80 counselors to describe their background academic experiences, the schools in which they work and the ways they spend their time at work. In addition, study participants were asked to describe their current counseling practices regarding computing, the extent to which computing was taught in their schools and their preferred methods of receiving professional learning. Understanding the ways in which school counselors interact with youth, parents, administrators, teacher colleagues and other stakeholders may assist K12 computing educators in their efforts to collaborate with counselors in a shared effort to recruit and retain youth in their courses. In this poster presentation the authors will a) frame the problem of counselor readiness to guide toward computing, b) describe the current experiences of school counselors with regard to computing education and career guidance, c) detail how counselors' current work assignments and backgrounds can inform professional learning and relationships in support of pathways to computing careers, and d) show sample messaging and resources computer science educators can use to inform counselors and other school professionals about opportunities for youth in computing.
Laptop Recommended. It is recommended that participants bring a laptop computer to this workshop

Keywords: counselor; computer science education; equity

DOI: http://dx.doi.org/10.1145/2839509.2850567

OnRamp to Parallel and Distributed Computing

Samantha S. Foley, *Univ. of Wisconsin-La Crosse*
Joshua Hursey, *Univ. of Wisconsin-La Crosse*
Daniel Koepke, *Univ. of Wisconsin-La Crosse*
Justin Ragatz, *Univ. of Wisconsin-La Crosse*
Jason Regina, University of Wyoming
Contact: ssfoley@cs.uwlax.edu

Computer science students must understand parallel and distributed computing (PDC) concepts to be effective computer scientists in the workforce, as reflected in the ACM Curriculum guidelines. Communities of CS educators are creating educational modules, and making parallel computer environments (PCEs) available to

educators who are integrating PDC concepts into their existing curriculum. Even with these resources there is a barrier to entry for students to use PCEs, namely the unfamiliar and complex system software ecosystem of modern PCEs. The OnRamp project lowers that barrier to entry for exploring PDC concepts on a variety of PCEs while also providing a path for students to learn how to be productive on the native PCE. OnRamp is designed to be a general-purpose web portal for supporting the exploration of PDC concepts that harnesses the existing educational resources created by the CS education community. It coaches students through interactive modules that teach them about PDC concepts and PCEs while allowing them to launch parallel applications from day one. As students become more comfortable with running parallel applications on PCEs, OnRamp transforms into a reference guide as they graduate to using the native PCE. This poster describes the motivation, design, and some motivating use cases for the project. We hope that this poster inspires parallel and distributed computing educators to use OnRamp in their courses.

Keywords: parallel and distributed computing; web portal; learning environment.

DOI: http://dx.doi.org/10.1145/2839509.2850568

Teaching Computational Thinking Through Bio-Design

Johanna Okerlund, *UNC Charlotte*
Orit Shaer, *Wellesley College*
Celine Latulipe, *UNC Charlotte*
Contact: jokerlun@uncc.edu

We are developing ways to teach computational thinking through interaction with tangible digital tools for synthetic biology. Inspired in part by Jeannette Wing's (Wing 2008) notion of the essence and pervasiveness of computational thinking, a growing community within SIGCSE is exploring tools that encourage computational thinking that don't use traditional computer science terminology or computer programming (Basawapna 2013, Daily 2014, Freeman 2014, Miller 2014, Wolz 2011). However, that community has not yet explored how to teach computational thinking through bio-design. Synthetic biology is a relatively new research field where scientists engineer living organisms to exhibit new behaviors. Biologists arrange and combine BioBricks, sequences of DNA that exhibit a behavior, sense an environmental factor, or indicate the beginning or end of a larger sequence. Bio-design not only incorporates traditional principles of computational thinking (abstraction, modularity, design/build/test), but also introduces new challenges that force the user to apply these principles in new ways. Interacting with biology can thus 1) help deepen understanding for people who are proficient in computational thinking, 2) provide a platform for introduction to computational thinking for the first time. Because most bio-design is performed in a wetlab by experienced biologists, we focus now on designing and developing low-barrier digital tools for interacting with biology. We present SynFlo (Okerlund 2016), a tangible interactive system for bio-design that mimics a real-life biological wetlab protocol and we discuss the properties of SynFlo and bio-design in general that encourage computational thinking.

Keywords: Computational thinking; Tangible interaction; Biology

DOI: http://dx.doi.org/10.1145/2839509.2850569

User Experience and Feedback on the RPI Homework Submission Server

Andrea Wong, *Rensselaer Polytechnic Institute*
Eric Tran, *Rensselaer Polytechnic Institute*
Joe Jung, *Rensselaer Polytechnic Institute*
Ben Shaw, *Rensselaer Polytechnic Institute*
Marina Espinoza, *Rensselaer Polytechnic Institute*
Beverly Sihsobhon, *Rensselaer Polytechnic Institute*
Melissa Lindquist, *Rensselaer Polytechnic Institute*
Samuel Breese, *Rensselaer Polytechnic Institute*
Matthew Peveler, *Rensselaer Polytechnic Institute*
Barbara Cutler, *Rensselaer Polytechnic Institute*
Contact: wonga6@rpi.edu

The Rensselaer Polytechnic Institute (RPI) Homework Submission Server is an ongoing, open-source project used to collect, compile, and automatically grade programming homework for students in our introductory and sophomore computer science classes. It allows viewing of homework, lab, test, and overall grades and late submissions and excused absences on homework. Our first hypothesis is that an electronic submission server is the preferred way for students to submit their coding homework because it provides immediate feedback about the correctness of their code and ensures fair, consistent grading since their code is compiled and run with the same test cases, on the same computer. Our second hypothesis is that students appreciate courses with a flexible policy for late submission of homework, allowing them to use a specific number of "late days" throughout the semester without penalty. We recently conducted a survey to test these hypotheses and will incorporate specific feedback from the students as we continue development and expand the server to more courses at RPI and other universities.

Keywords: CS1/2, Open Source

DOI: http://dx.doi.org/10.1145/2839509.2850570

Guiding Career Development Prior to Capstone Experiences

Deborah Knox, *The College of New Jersey*
Contact: knox@tcnj.edu

A new course, Computer Science Professional Development Seminar, engages students in reflective self-discovery and career planning, and also facilitates attainment of student outcomes assessment data for continual improvement of our program. Prior to requiring this half-semester course, students selected when to attend colloquia and various presentations. This strategy did not insure that students had a common exposure to topics. Though this course is titled as a professionalism course, this poster describes broader content coverage. Lifelong learning, ethical professionalism, and social media are included. The learning experiences in this course encourage students to establish an early commitment to their

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.

personal career development. Additionally, students are guided to increase their awareness of social issues and professional practice knowledge areas from CS2013. Targeted to the second-year (sophomore) population, this course helps connect the students with our departmental community and leads the students to better understand expectations and capstone opportunities. During this course, students receive mentorship by upper-class students, alumni, faculty, staff, and outside computing professionals. Since the course is required, students are afforded dedicated time to expand their understanding of professionalism, to learn about many opportunities for undergraduates, to reflect on case studies and on their intended career path, and to advance their preparations for the next step of their career.

This annual course has been offered four times. We strive to keep section enrollments at seminar level to facilitate discussion and to promote community. Some events with outside speakers require one common meeting of all sections. Course management strategies are offered.

Keywords: professionalism; career development; mentoring; lifelong learning; ethical professionalism

DOI: http://dx.doi.org/10.1145/2839509.2850571

Bigger Isn't Better When It Comes to Online Computer Science Teacher Communities

Mackenzie Leake, *Stanford University*
Colleen Lewis, *Harvey Mudd College*
Contact: mleake@cs.stanford.edu

Dozens of online communities have been developed to support high school computer science (CS) teachers by providing them with CS teaching resources. However, these sites have failed to meet teachers' needs and are widely underused. Despite this underutilization, organizations continue to create new communities with content that overlaps with the materials provided within existing communities. Our research explores the barriers to CS teachers' engagement with online resources and their attitudes toward online communities. We find that teachers are frustrated by the time and difficulty required to navigate the sites and find useful information. It appears the barriers to accessing these resources cannot be overcome by creating additional large, multipurpose communities. Even though it seems that having large communities would be valuable for increasing access to resources, our research indicates teachers prefer smaller, more specialized communities. We are eager to discuss ideas for designing new communities that provide more relevant content for teachers in a way that is easy for them to find.

Keywords: CS teachers; online communities; online resources

DOI: http://dx.doi.org/10.1145/2839509.2850573

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.

The Sol y Agua Project: Enhancing Middle School Education through Computing with an Emphasis on Simulation and Data Science

Ann Gates,*The University of Texas at El Paso*
Monika Akbar,*The University of Texas at El Paso*
Mary Kay Roy,*The University of Texas at El Paso*
Florencia Larsen,*The University of Texas at El Paso*
Ivonne Lopez,*The University of Texas at El Paso*
Christian Murga,*The University of Texas at El Paso*
Angel Ortega,*The University of Texas at El Paso*
Jesus Tellez,*The University of Texas at El Paso*
Rebecca Urbina,*The University of Texas at El Paso*
Contact: faramos2@miners.utep.edu

Data Science, the field of data extraction, manipulation, and analysis in order to derive knowledge or insight, is becoming an essential component of effective management and planning for industrial, research, scientific, and social communities. With the goal of engaging middle-school students in the important field of Data Science, the Sol y Agua project is developing a computer game based on regional environmental issues explored through data analysis. The Sol y Agua project will augment the learning environment of middle school students by incorporating simulation and data science to existing curricula in the form of an interactive educational game. The game has a theme of water sustainability and stewardship with an emphasis on information analysis, negotiation, and decision-making. The game will also immerse students in regional issues concerning biodiversity, sustainability, and the human impact on the environment. This video game will allow students to analyze and interpret data, and make informed decisions about water conservation and sustainability. Simulation and data visualization allow students to perform simulated, environmentally-conscious planning and virtual experiments. Using simulations allows students to easily and inexpensively gather data from the virtual environment. Data visualization allows us to quickly represent this data in new, meaningful, and engaging ways.

Keywords: Data visualization; Interactive Learning Environments; Simulation; Video-Game; Middle School Education

DOI: http://dx.doi.org/10.1145/2839509.2850574

An Expert System for the Prediction of Student Performance in an Initial Computer Science Course

Michael Kuehn, University of North Dakota
Jarred Estad, University of North Dakota
Jeremy Straub, University of North Dakota
Tom Stokke, University of North Dakota
Scott Kerlin, University of North Dakota
Contact: michaelkuehn12@gmail.com

There are many factors that play a part in how a student performs during a course. Some of these factors can be looked at in hindsight to help suggest why a student may or may not have done well in class. This poster assesses the potential of predicting student performance in an introductory computer science class using information about students' preparation, attitudes and study habits. An expert system has been utilized for this purpose. The expert system accepts data related to seven different categories of preparation, belief and attitude and, through the partial activation of multiple rules, predicts an outcome for each student on the post-test (which should correlate with and is used as a surrogate for the student's final course grade). This poster presents our findings and correlation between certain factors relevant to students' success. It shows how these factors can be used to predict a student's grade and discusses the significance of being able to do so. Thus far, the results from the first sample of data appear promising. The work has shown that certain groups of self-assessment questions have clearly out performed other groups in predicting final test grades. The use of expert system techniques to predict student grades can provide an insight on different factors that affect student performance. Knowledge of characteristics that may lead to poor performance allows instructors to potentially identify students who may be at risk of a low grade and need supplemental instruction, tutoring or other support at the outset of the semester.

Keywords: Computer science education; educational assessment; course outcome prediction; student learning

DOI: http://dx.doi.org/10.1145/2839509.2850576

A Web-based Environment for Developing and Utilizing Teaching Languages for Novice Computer Science Students

Benjamin J. Kruger, *Northeastern State University*
Richard Matzen, *Northeastern State University*
Contact: kruger@nsuok.edu

First year Computer Science students must face the dual challenge of learning to think algorithmically while simultaneously assimilating the complex and often unintuitive syntax of their first programming language. Mitigating this problem has been an active research area for decades, spawning a number of programming languages intended for students. While some such teaching languages have achieved a measure of popularity, both in the classroom and for general purpose programming, none has achieved ubiquity, which suggests there is no silver bullet. To this end, we present a suite of open source software including (1) a web-based toolkit for developing and modifying teaching languages using parsing expression grammars, (2) C Spot Run, a simple imperative teaching language built with the toolkit and informed by empirical research in intuitive syntax, and (3) a web-based development environment enabling the student to complete programming assignments directly from a web browser, eliminating complicated compiler installation and configuration. These artifacts are the first steps in an ongoing project intended to serve not only as a classroom tool, but also as a platform for the observational and experimental study of programming languages in classroom settings. By analyzing source code and metadata, common syntactic obstacles may be exposed and empirical studies of syntax can be verified in the classroom.

Keywords: Parsing Expression Grammars; PEG; PEG.js; Teaching Languages; Language Design; CS1; Open Source

DOI: http://dx.doi.org/10.1145/2839509.2850577

FunWithSound: A Music Composition and Synthesis Library for Processing

David Hovemeyer, York College of Pennsylvania
Contact: dhovemey@ycp.edu

FunWithSound is a music composition library for Processing which makes it easy for students in creative computing and introductory programming courses to create music compositions with a small amount of code. Building upon Processing's considerable strengths in the domain of visual art, FunWithSound extends Processing to support music creation.

Keywords: music; music synthesis; Processing

DOI: http://dx.doi.org/10.1145/2839509.2850530

Exploring the Role of Computer Science in the Liberal Arts

Gary R. Skuse, Rochester Institute of Technology
Daniel A. Walzer, University of Massachusetts, Lowell
Contact: grssbi@rit.edu

There is a growing body of evidence that indicates that many students would benefit from coursework in computer science regardless of their academic majors. While there are obvious advantages to learning computer science for students in the quantitative and analytical fields that comprise the STEM disciplines, the advantages to other students are less obvious. In order to investigate the impact of computer science principles and methods on students in the liberal arts we convened a workshop of faculty comprised of an equal number of participants from each discipline, i.e. computer science and the various liberal arts. We identified a clear need to better understand the computational needs of liberal arts students and faculty and we identified opportunities for computer scientists and liberal arts students and faculty to work together to better prepare students in both disciplines and better support faculty research in the liberal arts.

Keywords: Liberal Arts; disciplinary convergence; career nimbleness; multidisciplinary collaboration

DOI:http://dx.doi.org/10.1145/2839509.2850531

SIGCSE'16, March 2–5, 2016, Memphis, TN, USA.
ACM ISBN: 978-1-4503-3685-7/16/03.