Kinetic Art and Embedded Systems: A Natural Collaboration

Erik Brunvand
School of Computing
University of Utah
elb@cs.utah.edu

Paul Stout
Department of Art and Art History
University of Utah
paul.stout@gmail.com

ABSTRACT

We describe a cross-disciplinary collaborative course that pairs computer science and engineering (CSE) students with art students to engage in joint engineering design and creative studio projects. These projects combine embedded system design with sculpture to create kinetic art. We believe that this is a natural pairing of two disparate disciplines, and one that provides distinct educational benefits to both groups of students. In this paper we describe the course content, the collaborative process, the materials used in the class, and experience with a pilot version of the course taught in Fall 2009 at the University of Utah.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer Science Education

General Terms

Design, Experimentation

Keywords

Kinetic art, embedded systems, collaborative curricula, studio-based learning

1. INTRODUCTION

Jim Campbell is an artist living and working in San Francisco. According to a recent monograph on his work he is "...acknowledged as a masterful new media artist, adept at manipulating the base materials of electronics and computers into visual haiku of the information age." [9] He has been producing thought provoking and cutting-edge kinetic art for the past 20+ years. On his web site [7] he has a wry take on a "Formula for Computer Art", a snapshot of which is shown in Figure 1. The web-based version of Campbell's Formula is animated with many potential inputs and outputs scrolling by. The essential notion of Campbell's Formula is that as a broad category, kinetic art generally takes

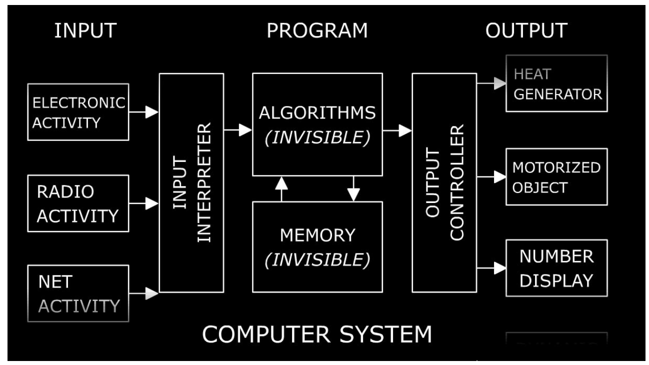

Figure 1: Jim Campbell's Formula for Computer Art [7]

some input from the real world (e.g., a sensor of some sort capturing environmental data in proximity to the artwork), processes it in an invisible way, and based on that input and processing, produces some output that is visible as the artwork itself.

Kinetic art contains moving parts or depends on motion, sound, or light for its effect. The kinetic aspect is often regulated using microcontrollers connected to motors, actuators, transducers, and sensors that enable the sculpture to move and react to its environment. But, distinct from other types of computer art, the computer itself is usually not visible in the artwork. It is a behind the scenes controller. An *embedded system* is a special-purpose computer system (microcontroller) designed to perform one or a few dedicated functions, often reacting to environmental sensors. It is embedded into a complete device including hardware and mechanical parts rather than being a separate computer system.

Kinetic art using embedded control is a marriage of art and technology. Artistic sensibility and creativity are required for concept and planning, and computer science and engineering skills are required to realize the artistic vision. In the project-based semester-length class we describe in this paper computer science and engineering (CSE) students work together with art students to build collaborative kinetic art pieces. Students explore interfacing of embedded systems with sensors and actuators of all sorts, along with real-time/interactive programming techniques and interrupt driven system design. They also explore physical and conceptual aspects of machine-making as a fine-art sculpture process.

SIGCSE'11, March 9–12, 2011, Dallas, Texas, USA.

The overall goal of the course is to enhance the educational experience for both groups of students. From the CSE side we encourage students to apply their technical skills in a context that is explicitly non-technical. This can allow students the freedom to try a wide range of approaches to a particular problem. In the process, engineers are exposed to a variety of aesthetic and creative concepts that would not normally be part of an engineering curriculum. Artists explore programming and engineering in a context that is more conducive to their learning style because it is directed at art-making. Both groups of students gain practical experience in design-thinking which is quite distinct from the computational thinking that is usually more associated with CSE. By requiring that the project groups include both engineers and artists, the students contribute to their own learning and creative growth through peer teaching. The multi-disciplinary teams allow both groups of students to experience a very different approach to problem solving because of their different fields. Learning to communicate across disciplines, and perhaps just as importantly respect each other's skills and contributions, is vitally important for successful collaboration. The overall result is that both student groups gain significant and unusual benefits that they can apply to a wide variety of aspects in their respective disciplines.

2. CURRICULUM

Our collaborative course is listed both as a CS course and as an Art course. The course is an upper division undergraduate course in both departments so we can assume that the CSE students are capable programmers and that the Art students are capable 3d artists. Students sign up for the course in their own discipline, but the classes meet together and the instructors team-teach all sessions. As we were planning the course this presented some challenges, mostly as we considered the amount of background material that each group of students might need in the other group's field of study. We considered breaking the class into two groups for the first part of the semester with the artists studying CS topics and the CSE students studying art. In the end we decided to keep all students together throughout the semester with the hope that students would rise to the occasion to contribute peer teaching to the curriculum when appropriate. We also considered the other extreme of pairing students at the beginning of the course, but were concerned that students from each discipline would rely too much on their counterpart and not attempt some of the material themselves.

The overall curriculum consists of review topics and individual assignments for the first four to five weeks of the semester, followed by a group project phase using a studio-based instruction model. This worked quite well in the pilot course. The topics covered include the following:

Computer Science and Engineering Topics
– Programming fundamentals
– Electronics fundamentals
– Input sensors (switches, potentiometers, resistive sensors including light, temperature, flex, etc., rangefinders, optical switches, etc.)
– Output actuators (servos, DC motors, stepper motors, LEDs, relays, switching transistors, etc.)
– Programming reactive systems
– External chip interfacing with protocols such as SPI
– Interrupt programming

Art and Design Topics
– Art history review of kinetic art
– Discussion of contemporary kinetic artists (Jim Campbell, Jack Dollhausen, Arthur Ganson, Rebecca Horn, Dan Rozin, Sabrina Raaf, Alan Rath, Peter Vogel, etc.)
– Formal elements of 3d art such as aesthetics, proportion, and balance
– Material studies (plastic, metal, paper, wood, etc.)
– Mechanical linkages and physical construction
– Concepts and meaning in art
– Artistic design process

Note that for each group of students, some of these topics will be review, and some will be brand new. For example, the first programming lab session is basic review for the CSE students, but may be the first time an art student has attempted to write a program of their own. From the other side, the fundamentals of 3d art is a review for the art students, but is perhaps the first time a CSE student has considered the ideas. While this has the potential to be "boring" to some of the students, we were pleasantly surprised in the pilot class that students seemed willing to take the reviews seriously, and help out during the hands-on activities.

Class proceeds with a set of individual assignments of gradually increasing complexity in both engineering and artistic aspects. These assignments introduce new programming concepts, give students experience with the range of sensors and actuators they can use, and practice building sculptural art pieces. After this phase, the students are partitioned into teams of three to four students where each team includes both engineers and artists. The teams proceed to work on two or three collaborative projects during the course of the semester. In our pilot offering of the course in 2009 we used teams of three and shuffled the teams after each project so that all students worked with a variety of different students throughout the semester.

An obvious question about the curriculum is how the subjects covered compare to a traditional CSE embedded systems course. Clearly because of the added art content, our kinetic art course cannot cover the same range of purely CSE material. In terms of strictly CSE content we cover interfacing to sensors and actuators in much more detail than our regular embedded systems course, but do not cover as much material related to assembly programming, advanced C programming, program optimization, or analysis of embedded programs.

The class culminates in a gallery show of the projects. This is a powerful motivator for the student groups to create interesting, thoughtful, and most importantly finished works by the end of the semester. A gallery show not only adds to the art students' portfolios, it can be an interesting resume line for the CSE students as well. We worked with the gallery in the Department of Art and Art History after the course in Fall 2009, and based on the success of that show are negotiating with other galleries in Salt Lake City for a show after this Fall's offering of the course.

2.1 Studio Art Educational Models

Pedagogical modes in engineering and fine art are typically quite different in both direct and subtle ways. One major difference is the use of studio-based instruction in the fine arts. Studio-based learning emphasizes collaboration, design-thinking, interaction, and learn-by-doing approaches. It forms the core of most fine art and design curricula [16, 21]. At the core of a studio-based curriculum are the ideas of exploring a multitude of design alternatives (design thinking) and the evaluation of those designs through peer group critiques (collaboration). These peer critiques serve a very different role for the student from graded assignments or even from engineering design reviews. Because students are discussing, evaluating, and critiquing each other's work, this can create a much more collaborative, supportive, and less hierarchical learning environment than traditional engineering courses with lectures and individually completed lab assignments.

Our course includes peer design critiques at both formative and completed project phases. This is natural for the art students, but can be intimidating to the CSE students at first. We found that it takes a few critique sessions for the CSE students to feel comfortable with the process. Because the designs are not "regular" CSE projects, we feel that this environment allows the CSE students to more comfortably adjust to the critique process. That is, because the subject of the critiques is (for the most part) art, their CSE skills are not necessarily on the line.

Although studio-based curricula are the norm in fine arts and design fields (such as industrial design, architecture, graphic design, etc.), they are not common in engineering. Recently there have been some intriguing studies on how a more collaborative [4, 35, 27, 10] or studio-type curriculum [21, 6, 19, 20, 17] might be adapted for CSE. We find these approaches very interesting from a broader CSE point of view, but view them as essential for our proposed kinetic art curriculum. One of our ongoing curriculum development challenges is to fit the programming, computing, and engineering construction course content into this type of exploratory, collaborative studio environment. A longer term goal is to contribute to the growing body of knowledge about how a studio model meshes with a broader CSE curriculum.

2.2 Enhancing Creativity

Computer scientists and computer engineers who exhibit high levels of creativity and interdisciplinary design vision are highly valued in their communities, and are often identified with leadership roles in research, development, and education. Creativity is clearly a tremendous asset when faced with a challenging problem, and students who can work across disciplines are a natural fit for many areas of CSE studies. However, students with this powerful combination of traits are seen as somewhat rare in CSE. One of the exciting ancillary benefits of this kinetic art and embedded systems course is the potential to enhance the creativity of our CSE students.

In this course, technical students are encouraged to apply their technical skills in a context that is explicitly non-technical. This not only forces students to think about problems and solutions in a very different way, but also gives them an environment where they can feel able to propose and explore solutions where aesthetics are a primary goal, not engineering costs and practicalities. This freedom to explore "impractical" solutions is, we believe, a powerful creativity enhancer. One interesting model of creativity defines enhanced creative problem solving in the following way: "Creativity is demonstrated by the generation of many potential solutions instead of gravitating quickly toward a single and (usually) familiar solution that is not necessarily the optimal one." [11]

Thinking outside the normal scope of engineering projects can have a profound impact on how engineers and computer scientists approach all their designs. Students who have more creative, artistic design in their engineering vocabulary should be able to design better human interfaces, make systems more usable, extendable, and testable, and be more willing to take novel and unconventional approaches. We think of this as extending the palette available when considering the design of a new system, whether that system is software, hardware, mechanical, or a combination of all. There are a number of researchers who report that interdisciplinary collaborations are a powerful catalyst for creativity [27, 8, 4, 35].

Another aspect of creative problem solving is embodied in the actual construction of a physical artifact. The root of the word "creativity" itself implies the creation of something. There is evidence that creativity is enhanced by engaging in studio courses where the projects involve physical artifacts [12, 10, 32, 18].

3. COURSE FACILITIES

The primary facilities required for a course of this type are studio space that supports the exploration and construction of moderately complex 3d assemblages, a variety of sensors and actuators that the students can use, and a microcontroller platform for control.

In terms of studio space our class is able to use the 3d design studios in the Department of Art and Art History. This includes separate wood and metal shops, as well as open studio space for project construction. In terms of sensors and actuators, we received a small grant from the University of Utah Teaching Committee to support our pilot class in Fall 2009 and used that, in part, to stock our lab with initial supplies. A modest class fee for subsequent classes will be used to restock and acquire a variety of components for the lab. These components (servos, motors, LEDs, light sensors, resistors, etc.) are for the most part quite inexpensive ranging from a few pennies to a few dollars each.

For the microcontroller platform there are a variety of choices available. The choice of a controller can have a large impact on the class because the control of the kinetic art is critical, and different controllers have quite different features and programming environments. We have chosen to use the Arduino embedded controller and associated programming environment [2]. The Arduino is an open-source electronics prototyping platform based on an Atmel AVR 8-bit microcontroller. The hardware is open-source in the sense that all details of the controller board are made freely available so that users can build their own if they choose. The software integrated development environment (IDE) for programming the Arduino hardware is also called Arduino, and is a set of C-language functions designed to be relatively easy for any user to quickly write programs that use the Arduino hardware to sense and control the physical world.

Although the Arduino contains many interesting I/O properties and features, it is quite affordable. The basic board we are using is the Arduino Duemilanove which has 14 digital I/Os, 6 Analog inputs, 1 UART, and a variety of other features on a 2.7 X 2.1 inch board and costs around $30 (See Figure 2). The development and programming environment

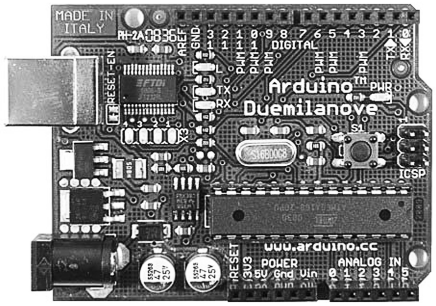

Figure 2: An Arduino Duemilanove board which is based on the AVR ATmega 328p microcontroller.

is open source and available free of charge in PC, Mac, and Linux versions. The Arduino programming IDE is really C/C++, but with a set of useful functions predefined for the user. The back-end of the IDE is the gcc compiler [15] which means that advanced users can use the full C/C++ programming language, and access all the internal features of the Atmel chips. Because Arduino hardware is programmed through the USB port of the host computer (acting as a serial interface), no additional programming hardware needs to be purchased.

Although the Arduino is a relative newcomer to the microcontroller world (compared to PIC and Basic Stamp, for example), it already has a wide following which results in many informative web sites that students can access [13, 14, 25]. There are also a number of books that describe using Arduino for controlling and sensing the physical world [29, 22, 5].

4. PILOT COURSE

A pilot version of the course was taught by the authors in Fall semester 2009. The enrollment was fairly small with six CSE and three art students participating. All aspects of the curriculum in Section 2 were covered, and a total of three group projects were completed by each of the three interdisciplinary teams. Because of the small class size, we were able to shuffle the teams such that every student worked on a team with every other student at some time during the semester, and all teams consisted of both artists and engineers. The course culminated with a gallery show in the Gittins Gallery in the Department of Art and Art History in January 2010. The show consisted of all nine group project pieces and was widely viewed as a successful show by other faculty in both departments (see Figure 3 and http://www.eng.utah.edu/˜cs5968/pictures).

Some details of a few of the final kinetic art projects from Fall 2009 are:

***Relic*:** This large floor-standing piece (seen in Figure 3) was constructed of wood and metal with stepper motors, light bulbs, and electronic (Arduino) control. Standing in front of the piece the viewer sees a triangular themed structure with a dim light in the center casting a shadow of some gears on the wall behind. There are two metal plates on either side of the piece with hand prints on them inviting the viewer to place their hands there. The piece sends a very low voltage signal through those plates to sense the capacitance of the person and increases the light output when someone is touching the plates. As more people join hands and make a human chain between the plates, the light grows brighter and the gears turn casting a dynamic shadow on the wall. Drone-like sound effects are also played during this process. The piece is meant to evoke the idea of a mystical relic that responds to cooperation by the viewers. The more viewers join the human chain, the brighter and more dynamic the relic becomes.

***Cars*:** In this piece a large eight foot square white board was set up as a table. On the table are two small remote control cars, stripped to their chassis, and each dragging a white board marker. The cars are controlled by Arduinos hidden under the table. The turning and direction of the cars is changed in response to characters received on internet RSS feeds so that the patterns drawn on the white boards are in some way a physical representation of the data coming across that internet feed. By placing paper under the cars a permanent record of the drawing can be made.

***Flowers*:** In this piece eight kinetic flowers (each around 18" in diameter) were constructed from plastic, foam, paint, and servos controlled by an Arduino (see Figure 3). At the center of each flower is a light sensor. When left on their own the petals of the flowers move slowly up and down in a random fashion that looks calm. When a viewer moves close enough to change the amount of light on the sensor however, the flowers become agitated and exhibit a variety of different frightened or angry behaviors until the viewer backs away.

***Windchime*:** This piece consists of a wind chime constructed from copper tubing for the chimes, and a decorative metal housing as the support. At the center of the chimes is a metal striker. As the striker hits each of the chimes, a switch connection is made and a different set of LEDs in the main housing illuminates. A gentle breeze provides a constantly changing set of sounds and colors.

The student evaluations of the course were very positive with numeric scores above the department and college averages. The most striking thing about the comments was the degree to which students in each demographic group were positive about the experience of working on such wildly multi-disciplinary teams. Some representative student comments include: "It was really great to learn how to work with other students (art, cs, ece) like on the job." "I really enjoyed the open interaction that all the students had with each other and with the instructors." "I learned a great deal about electronics and embedded programming that I had not learned previously in my engineering courses. For me, it put into practice and solidified what we covered in the earlier engineering courses."

We are encouraged that the enrollment in our Fall 2010 course is more than double our Fall 2009 course. The course web page is available at http://www.eng.utah.edu/˜cs5968. One of the main lessons we learned in the pilot course was that without individual practice in all aspects of the kinetic

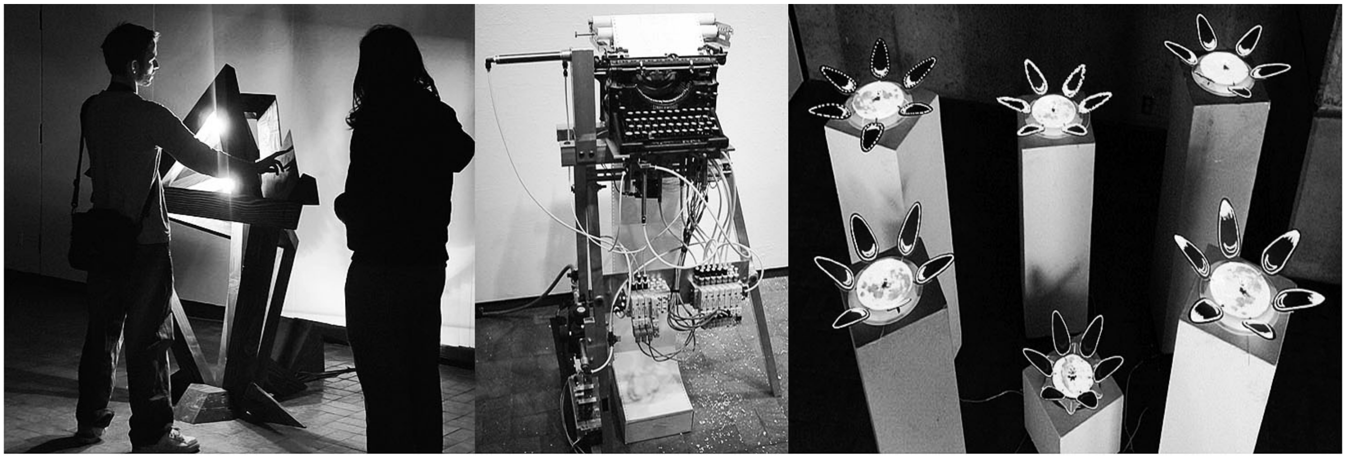

Figure 3: Three of the nine finished projects from *Invisible Logic*, the gallery show from Fall 2009. From the left they are *Relic* (wood, metal, light bulbs, motors, electronic control), *Underwood 1910* (metal, typewriter, pneumatic actuators, electronic control), and *Flowers* (plastic, paint, servos, electronic control)

art design process, multi-disciplinary teams tended to naturally slip into predefined engineering/art roles. In our second offering we are adding more extensive individual projects before forming teams, and will monitor the teams carefully to judge team participation in all aspects of the projects.

Another lesson is that the more hands-on practice with the materials, the better. We have a variety of sensors and actuators available for the students, but in the end most art pieces used a small subset of them, mainly servos, LEDs and light sensors. This is partly because those components already provide a great deal of flexibility in how they can be integrated into the kinetic art, but it likely also because they were the components most prominently featured in our initial hands-on labs. For our current offering of the course we plan to increase the number and type of components used in the preliminary labs to see if that will result in a wider range of components being used in the final projects.

One lesson that may translate to many other types of CSE courses is how valuable the process can be of building small throw-away non-serious prototypes. Students in the Fall 2009 semester commented on how liberating it was to sketch rough ideas and try things out on breadboards in class just to see how the components behaved without worrying about making every test a completed assignment. As described earlier, one intriguing definition of creativity is how many designs are explored before a final solution is settled upon. A kinetic art project, along with the ongoing design critiques in the studio class model, is a perfect opportunity to explore creativity in this way.

5. RELATED WORK

The general area of collaboration between arts and technology is quite rich, but many educational collaborations are targeted specifically at finding ways to introduce technology to artists, designers, and other non-technical or younger audiences (e.g.,affordable Alice [1, 23], Scratch [33, 26], Processing [30, 31], etc.). We have found only a few examples of using arts to enhance engineering. Some of those have already been referenced in this paper and include exploration of how interdisciplinary collaboration in general enhances engineering problem solving [4, 35], how studio-based courses make sense as a model for computing courses [12, 21, 6], and how the process of making tangible artifacts enhances engineering problem solving [10, 32, 18, 3].

The program most directly related to our Kinetic Art and Embedded Systems course is the successful Artbotics program at UMass Lowell [24, 28, 34], initiated in 2006. This is quite a similar program in some respects. Its main activity involves a curriculum using art and robotics (similar to kinetic art, although it is not described in that way) to help attract new students to computing disciplines. They are also connected with a local museum that organizes shows of the student's work. There are many small, but important, differences between Artbotics and our proposed course, but one major difference is that Artbotics is primarily an after school program and summer camp for high school students. Artbotics is also somewhat more focused on the robotics aspects of the projects than the fine-art aspects of the projects although this is admittedly somewhat subjective.

6. CONCLUSIONS AND FUTURE DIRECTIONS

Our collaborative course builds on the powerful connection between embedded control and kinetic art. This pairing seems like a natural fit, and one with high potential for intriguing results. Engineers are rarely taught to think about artistic, conceptual, and aesthetic outcomes, and artists are not usually taught to think about engineering issues in creating an artistic artifact. The studio model is an intriguing model for more general CSE education, but it is perhaps best experienced in a true studio course. A focus on design thinking also seems to us to be a natural complement to computational thinking.

There are many opportunities for this sort of collaboration. Many media in the fine arts require the equivalent of engineering problem solving to master. Kinetic art is the most obvious (and thus is the choice for this course), but sculpture, printmaking, photo, new media, video, installation art, interactive art, and digital media all have aspects that would be amenable to collaboration of this sort.

Enhanced collaboration between these colleges that are usually thought of as quite distant could have a profound impact on how both engineers and artists are educated. Our broader goals are to use this as a building block for further connections between the College of Engineering and the College of Fine Arts at the University of Utah, and to serve as a model that could be adopted at any number of other educational institutions. Further connections could engage a much wider range of engineering and fine arts students, and explore further fascinating synergies in creative problem solving in both engineering and fine arts.

7. ACKNOWLEDGMENTS

We thank the University of Utah Teaching Committee for their support of our pilot course in Fall 2009.

8. REFERENCES

[1] The Alice web page. http://www.alice.org.

[2] Arduino home page. http://www.arduino.cc.

[3] D. Avrahami and S. E. Hudson. Forming interactivity: a tool for rapid prototyping of physical interactive products. In *DIS '02: Designing Interactive Systems*, pages 141–146, London, England, 2002.

[4] E. E. Ayiter, S. S. Balcisoy, M. Germen, and S. Artut. Propagating collaboration: an instructional methodology for artists and engineers. In *SIGCHI C&C '07*, Washington, DC, USA, 2007.

[5] M. Banzi. *Getting Started with Arduino (Make: Projects)*. Make Books, 2008.

[6] L. J. Barker, K. Garvin-Doxas, and E. Roberts. What can computer science learn from a fine arts approach to teaching? In *SIGCSE '05*, St. Louis, Missouri, USA, 2005.

[7] J. Campbell. Artist's web site. http://www.himcampbell.tv.

[8] L. Candy and E. Edmonds. *Explorations in art and technology*. Springer-Verlag, London, UK, 2002.

[9] S. Dietz, editor. *Jim Campbell: Material Light*. Hatje Cantz, 2010.

[10] E. Y.-L. Do and M. D. Gross. Environments for creativity: a lab for making things. In *SIGCHI C&C '07*, pages 27–36, Washington, DC, USA, 2007.

[11] E. Edmonds and L. Candy. Creativity, art practice, and knowledge. *Commun. ACM*, 45(10):91–95, 2002.

[12] E. A. Edmonds, A. Weakley, L. Candy, M. Fell, R. Knott, and S. Pauletto. The studio as laboratory: combining creative practice and digital technology research. *Int. J. Hum.-Comput. Stud.*, 63(4-5):452–481, 2005.

[13] The world famous index of arduino and freeduino knowledge. http://www.freeduino.org.

[14] L. Fried. Arduino tutorial. http://www.ladyada.net/learn/arduino.

[15] Gcc, the GNU compiler collection. http://gcc.gnu.org.

[16] L. N. Green and E. Bonollo. Studio-based teaching: history and advantages in the teaching of design. *World Transactions on Eng. and Tech. Edu.*, 2(2):269–272, 2003.

[17] D. Hendrix, L. Myneni, H. Narayanan, and M. Ross. Implementing studio-based learning in CS2. In *SIGCSE '10*, Milwaukee, Wisconsin, USA, 2010.

[18] S. E. Hudson and J. Mankoff. Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape. In *UIST '06*, Montreux, Switzerland, 2006.

[19] C. Hundhausen, A. Agrawal, D. Fairbrother, and M. Trevisan. Integrating pedagogical code reviews into a CS1 course: an empirical study. In *SIGCSE '09*, Chattanooga, TN, USA, 2009.

[20] C. Hundhausen, A. Agrawal, D. Fairbrother, and M. Trevisan. Does studio-based instruction work in CS1?: an empirical comparison with a traditional approach. In *SIGCSE '10*, Milwaukee, Wisconsin, USA, 2010.

[21] C. D. Hundhausen, N. H. Narayanan, and M. E. Crosby. Exploring studio-based instructional models for computing education. *SIGCSE Bull.*, 40(1), 2008.

[22] T. Igoe. *Physical Computing: Sensing and Controlling the Physical World with Computers*. Course Technology PTR, 2004.

[23] C. Kelleher, R. Pausch, and S. Kiesler. Storytelling Alice motivates middle school girls to learn computer programming. In *CHI '07*, pages 1455–1464, San Jose, California, USA, 2007.

[24] H. J. Kim, D. Coluntino, F. G. Martin, L. Silka, and H. A. Yanco. Artbotics: community-based collaborative art and technology education. In *SIGGRAPH '07: Educators Program*, San Diego, California, 2007.

[25] T. Kurt. Todbot blog. http://todbot.com/blog/.

[26] D. J. Malan and H. H. Leitner. Scratch for budding computer scientists. In *SIGCSE '07*, Covington, Kentucky, USA, 2007.

[27] L. Mamykina, L. Candy, and E. Edmonds. Collaborative creativity. *Commun. ACM*, 45(10):96–99, 2002.

[28] F. Martin, H. J. Kim, L. Silka, H. Yanco, and D. Coluntino. Artbotics: Challenges and opportunities for multi-disciplinary, community-based learning in computer science, robotics, and art. In *Workshop on Research in Robots for Education*, Robotics Science and Systems conference, June 2007.

[29] J. Noble. *Programming Interactivity: A Designer's Guide to Processing, Arduino, and openFrameworks*. O'Reilly Media, Inc., 2009.

[30] Processing web site. http://processing.org/.

[31] C. Reas, B. Fry, and J. Madea. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2007.

[32] O. Shaer, M. s. Horn, and R. j. k. Jacob. Tangible user interface laboratory: Teaching tangible interaction design in practice. *Artif. Intell. Eng. Des. Anal. Manuf.*, 23(3):251–261, 2009.

[33] The Lifelong Kindergarden Group at MIT. The Scratch web page. http://scratch.mit.edu/.

[34] H. Yanco, H. J. Kim, F. G. Martin, and L. Silka. Artbotics: Combining art and robotics to broaden participation in computing. In *AAAI: Resources for AI Education*, Stanford, CA, March 2006.

[35] Y. Zhang and L. Candy. An in-depth case study of art-technology collaboration. In *SIGCHI C&C '07*, Washington, DC, USA, 2007.